Ottobre/Novembre 1990, Anno V n. 5

La rivista per utenti di C-64/128 ed Amiga

COMMODORE GAZETTE

smau

AMIGA 3000:

IL SISTEMA OPERATIVO 2.0

SMAU e IBTS:

TUTTE LE NOVITÀ COMMODORE

Prove software Amiga:

AMOS E POWERWINDOWS 2.5

KCS Power PC Board:

COMPATIBILITÀ IBM XT
PER AMIGA 500

Programmare l'Amiga:

- I PROCESSI DELL'AMIGADOS
- LA GESTIONE DEL JOYSTICK IN C

Commodore 64/128:

IL NUMERO DI DISPOSITIVO

Sped. in abb. post. Gr. III/70 – Lire 8.000

Sommario

OTT./NOVEMBRE 1990
Anno V, N.5

ARTICOLI

RUBRICHE

è una pubblicazione

Direttore responsabile: Massimiliano M. Lisa
Redazione: Dario Tonani, Nicolò Fontana Rava
Collaborazione editoriale: Fabio Rossi, Avelino De Sabbata, Fulvio Piccioli, Alfredo Prochet, Michael Moosleitner, Stefano Franzoto, Giorgio Bicego
Corrispondenti USA: William S. Freilich, Daniela D. Freilich
Collaborazione editoriale USA: Matthew Leeds, Ervin Bobo, Eugene P. Mortimore, Morton A. Kevelson. *AmigaWorld*: Bill Catchings, Mark L. Van Name, David T. McClennan. *Run*: Mark Jordan
Segretaria di redazione: Giovanna Varia
Impaginazione e grafica: Andrea De Michelis
Fotografie: A.&D., Patricia Leeds
Disegni: M.P., G. F.

Direzione, Redazione, Amministrazione: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano
Fotocomposizione: IHT Gruppo Editoriale S.r.l. - Divisione grafica
Fotolito: Colour Separation Trust S.r.l. - Via Melchiorre Gioia, 61 - 20124 Milano
Stampa: OFSA S.p.A. - Via Giacomo Puccini, 64 - 20080 Casarile (MI)
Distribuzione per l'Italia: Messaggerie Periodici S.p.A. - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467545
Distribuzione per l'estero: A.I.E. S.p.A. - Via Gadames, 89 - 20151 Milano - Tel. 02/3012200

Pubblicità: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-799492-792612-794122 - Fax 02/784021 - Telex 334261 IHT I - Agente pubblicitario: Aldo Pagano Pagano

Abbonamenti: IHT Gruppo Editoriale Servizio Abbonati - Via Monte Napoleone, 9 - 20121 Milano. Linea per registrazione abbonamenti: 02/794181 799492-792612-794122
Costo abbonamenti: Italia 12 numeri L. 96.000 - 24 numeri L. 192.000 - 36 numeri L. 288.000
Estero: Europa L. 150.000 (10 numeri) Americhe, Asia... L. 200.000 (10 numeri). I versamenti devono essere indirizzati a: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano mediante emissione di assegno bancario o vaglia postale

Arretrati: Ogni numero arretrato: L. 16.000 (sped. compresa)

Autorizzazione alla pubblicazione: Tribunale di Milano n. 623 del 21/12/85. Periodico mensile. Sped. in abb. post. gr. III/70. ISSN: 0394-6991
La IHT Gruppo Editoriale è iscritta nel Registro Nazionale della Stampa al n. 2148 vol. 22 foglio 377 in data 5/6/1987

Associato alla U.S.P.I. (Unione Stampa Periodica Italiana)

NOTE EDITORIALI

La nostra videocassetta ***Computerarte, Computergrafica e Animazioni Vol. 1 (The AmigaWorld Animation Video Vol. 1)*** ha ottenuto un lusinghiero successo di pubblico.

Per quei lettori che non l'avessero acquistata, si tratta di una videocassetta VHS con sonoro stereo hi-fi, che raccoglie circa 60 minuti di animazioni, e opere di computerarte e di computergrafica in genere, realizzate dai più grandi talenti internazionali della comunità Amiga. È un prodotto audiovisivo davvero affascinante, dedicato a una fascia di utenti ampissima: non solo quindi ai possessori di una Amiga.

Se possedete un Amiga, ovviamente il sapere che quanto vedete è stato realizzato con la macchina che avete in casa è decisamente stimolante alla creazione di opere personali. Ma anche se non disponete di un Amiga, e siete utenti C-64, IBM, Atari, Mac... troverete questa cassetta davvero affascinante. L'ha trovata affascinante anche chi di computer non sa niente.

Se nelle edicole della vostra città non è disponibile, potete comunque ordinarla attraverso il modulo pubblicato nella pubblicità che troverete nelle pagine di questo stesso numero. Oppure potete cercarla in libreria.

Si tratta di un acquisto del quale non vi pentirete. Vi posso dare la mia assicurazione personale.

Per tutti i nostri lettori interessati alla programmazione in Assembly, è finalmente disponibile in libreria ***Il Manuale dell'Hardware dell'Amiga.***

Per chi invece avesse interessi rivolti all'astronomia, è appena uscito in libreria un nuovo titolo appartenente alla collana Tempus – Scienza e Tecnologia edita dalla IHT. Si tratta de ***L'Universo del giovedì*** di Marcia Bartusiak (ISBN 88-7803-015-5, 39.900 lire, 344 pagine, distribuito dalla RCS Rizzoli Libri, edito dalla IHT Gruppo Editoriale). È un libro davvero interessante e affascinante che negli Stati Uniti è stato definito dal *New York Times* uno dei migliori testi scientifici dell'anno. L'autrice ha anche ricevuto un premio dall'American Institute of Phisics in riconoscimento della sua attività nel settore della divulgazione scientifica. È un testo indicato per tutti gli appassionati di astronomia, dai neofiti ai cultori di lunga data, ma anche per studenti, professori e astronomi.

Nasce Arte in Video

Questo mese la nostra casa editrice ha tenuto a battesimo una sua nuova creatura: *Arte in Video.* Si tratta di un mensile venduto in edicola composto di una videocassetta d'arte, che ogni numero è dedicato a un argomento diverso. La videocassetta è accompagnata anche da una rivista.

Questo mese l'appuntamento è con ***Vincent van Gogh***. Attraverso questo raffinato documentario si arriva a conoscere da vicino i dipinti e il dramma esistenziale di van Gogh. Oltre a ricchissime parti filmate dei luoghi nei quali l'artista è vissuto, la macchina da presa lascia parlare molto anche le opere e i loro particolari, suscitando nello spettatore emozioni intense e davvero uniche. Il documentario appartiene a una collezione di opere realizzate dal Metropolitan Museum of Art di New York, uno dei musei più importanti e prestigiosi del mondo.

Correte quindi in edicola a chiedere *Arte in Video/Vincent van Gogh*. Tenete presente che circa un mese dopo la loro uscita in edicola le videocassette escono poi in libreria. Se in edicola fosse esaurita, potete quindi trovarla in libreria (distribuzione in libreria: RCS Rizzoli Libri).

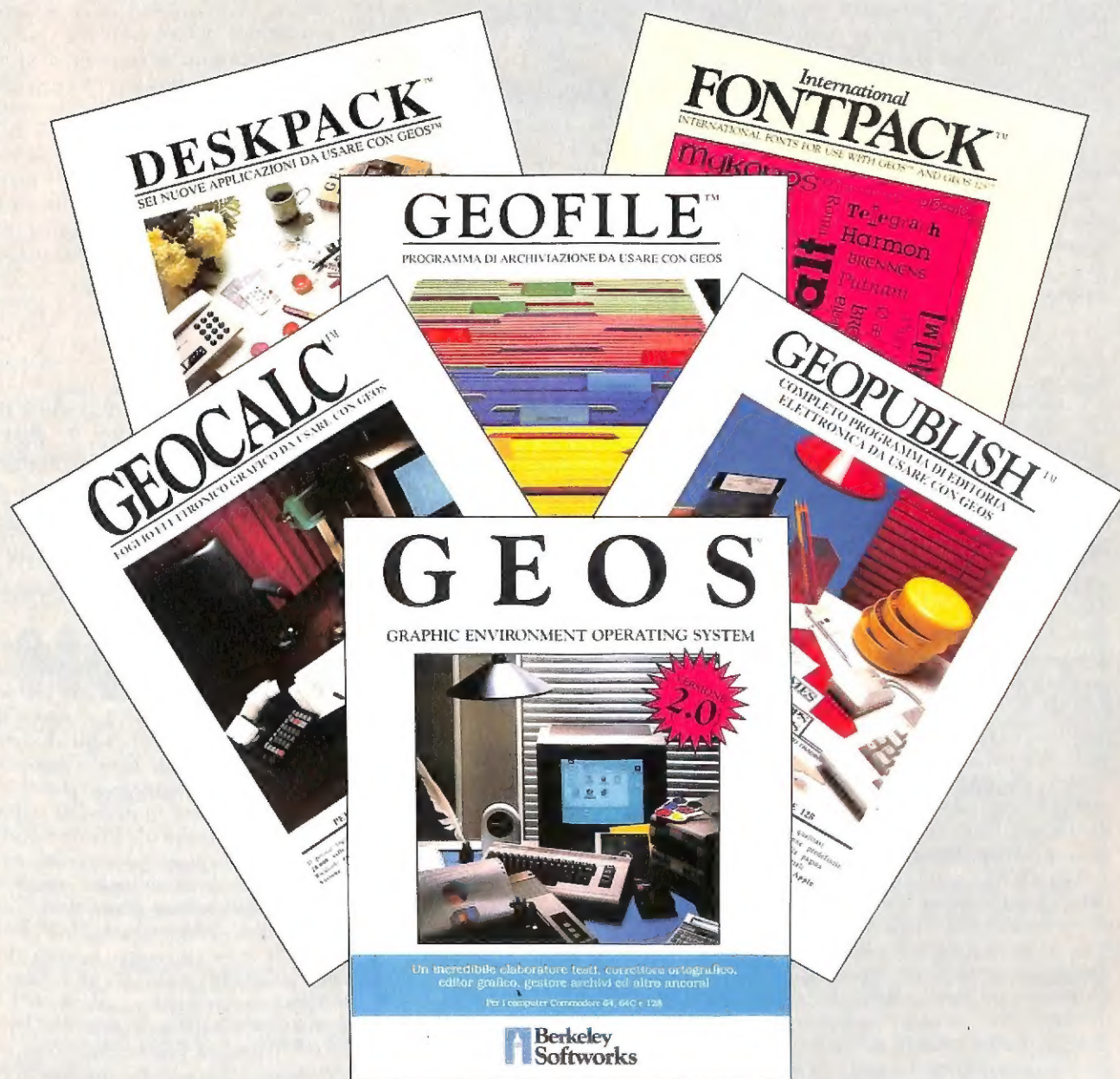
Intorno al 10 di dicembre l'appuntamento in edicola sarà invece con il numero 2 di *Arte in Video*, che sarà intitolato: *I capolavori del Metropolitan*. In questo documentario il direttore del Metropolitan Museum of Art, ci guida in una visita privata ai più bei capolavori del celebre museo newyorkese.

La videocassetta vi permette un incontro davvero intimo con l'arte che nessun altro media è in grado di realizzare. *Arte in Video* rappresenta quindi un'iniziativa editoriale innovativa che non potete perdere.

Nelle nostre videocassette non ci sono lunghe trattazioni cattedratiche. Sono le immagini a parlare, a parlare con il vostro cuore.

M.L.

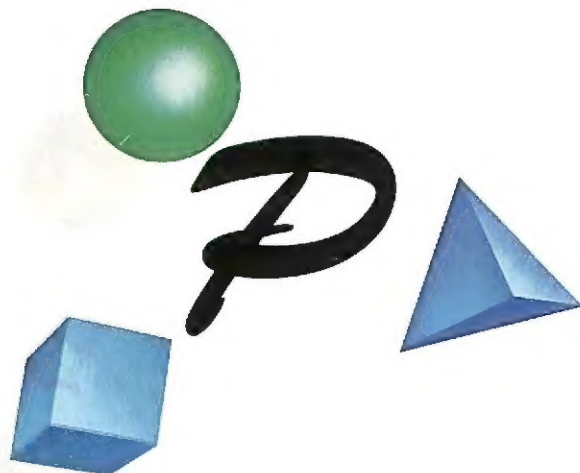
G E O S 2.0

ABBIAMO QUALCOSA DA PROPORVI SETTE VOLTE MEGLIO DI GEOS:

COD.	PRODOTTO	PREZZO AL PUBBLICO
8744	GEOS 2.0	L. 99.000
7950	GEOFILE	L. 69.000
7949	GEOCALC	L. 69.000
7948	GEOPUBLISH	L. 69.000
5848	DESKPACK	L. 49.000
7785	FONTPACK	L. 69.000

Disponibile presso i negozi SOFT CENTER e i migliori rivenditori.

Leader Distribuzione srl - Via Mazzini 15 - 21020 Casciago (VA)

LA POSTA DELLA GAZETTE

LA VOCE DEI NOSTRI LETTORI

PAROLE IN LIBERTÀ

Il motivo della mia lettera è presto detto: sono stufo di come vengono trattati i clienti Commodore non solo dalla casa madre, ma da tutti i produttori in genere.

Innanzitutto, vorrei parlare della scheda Superotto Hardital che ho appena rivenduto: è ben realizzata, assemblata con cognizione di causa, con tanti condensatori di filtro come poche altre! Eppure ho avuto dei problemi con un hard disk A2090A da 20 MB, che non voleva condividere la macchina. La ditta Computer Center, alias Hardital, non è molto esplicita con questi problemi, e questo a discapito di chi legge un semplice annuncio pubblicitario.

Allo SMAU non un 2000, qualche 500 relegato a semplice game machine, niente CDTV, la stampante a getto d'inchiostro era solo nel retostand per pochi eletti... Forse qualcuno ha letto il nuovo opuscolo Commodore che veniva regalato allo SMAU: il modulatore A510 ha l'ingresso audio miscelato infilato nell'uscita RF, la scheda 2288AT è la stessa foto della 2088XT, l'A2091 è invece un 2094, ma non è neppure citata l'importantissima proprietà di connettere 2 MB di RAM. Ma non è finita qui. Sempre allo SMAU mi sono rivolto a un addetto Commodore, e mi ha consigliato vivamente di rivolgermi alle ditte esterne, perché gli HD sono più veloci, potenti e a buon mercato. Ogni commento è sprecato. Arriviamo ad oggi, con l'acquisto della MPS1224C trovata a un prezzo ottimo: 760.000 lire!

La competenza e la cordialità del punto Commodore Righi Elettronica di Uggiate Trevano (CO) è esemplare, purtroppo però: quale drive si utilizza per questa stampante? Inutili i tentativi con NEC, Epson (Q, Old, X!), e con tutte le combinazioni di emulazione. Niente da fare, la grafica risulta "shiftata" a destra con rientro o peggio! Poi, grazie al Commodore Point suddetto, mi procuro il driver Epson-JX80, che pare vada bene. Ma che fare per i colori troppo scuri? Inoltre, questo driver non è presente nel direttorio di *WB 1.3*, e come faccio io a procurarmelo? Naturalmente non esiste alcuna istruzione nel manuale, esempio di traduzione classica. Dai due volumi in lingua tedesca, a uno malamente stampato, misero e inutile, in italiano...

Massimo Moltoni
Como

Indirizzate tutta la corrispondenza per la rivista a:

COMMODORE GAZETTE
La posta della Gazette
Via Monte Napoleone, 9
20121 Milano

Preghiamo i lettori di essere concisi e concreti, per darci modo di rispondere al più grande numero possibile di lettere. La redazione si riserva comunque il diritto di sintetizzare le lettere troppo lunghe.

LA COPIA E LO SCONTRINO

Vi scrivo questa missiva per mettere in luce ancora una volta il problema dei negozi che dovrebbero vendere software originale e che, invece di garantire la validità e l'originalità del software, compiono vere e proprie truffe ai danni degli utenti. Mi spiego. Circa un mese fa decisi di acquistare *Drakken* originale in un negozio di Milano, ma, poiché il prezzo era abbastanza alto, ripiegai, stupidamente, sulla versione pirata costituita da due dischi, come l'originale, ben felice di aver risparmiato soldi (ho pagato 20.000 lire, 10.000 lire a dischetto). Arrivato a casa, inserii il dischetto e cominciai a giocare. Dopo un po' di partite, non essendo riuscito a combinare niente, mi domandai se il gioco fosse a posto o meno, ma alla fine prevalse l'idea che, essendo un RPG, difficilmente avrei appreso lo schema di gioco senza istruzioni.

Dopo tre settimane di vari tentativi, tornai al negozio per acquistare un altro programma, ma sfogliando la lista, notai la dicitura "3 dischi" per *Drakken*. Abbastanza arrabbiato per questo chiesi al negoziante di cam-

biarmi gioco, ma senza risultato, adducendo come scusa che per cambiare i dischetti bisognava avere lo scontrino fiscale, da me buttato via. Tra l'altro ho notato che molti altri game vengono, per così dire, "spezzati" come *TV Sport Basket*, venduto su tre dischi invece che su due, solo per far pagare di più agli ingenui clienti. Oltre, quindi, a protestare per questo fatto, a pentirmi per la mia ingenuità, e a rammaricarmi del fatto che simili persone guadagnino all'ombra della legge, mi chiedo se effettivamente lo scontrino sia valido ai fini della restituzione di una cosa illegale e che cosa fa la legge per punire questa gente.

Andrea
Milano

EVVIVA IL VIDEO

Sono un ragazzo di 28 anni, appassionato di videoregistrazione; ora avendo da poco acquistato il secondo videoregistratore e possedendo ormai da alcuni anni una videocamera, ho sentito la necessità di perfezionarmi e ho acquistato, alcuni mesi fa, un Amiga 500, completo di secondo drive ed espansione interna da 512K. L'acquisto di tale apparecchiatura è e sarà riservato esclusivamente ad arrangiamenti di animazione (*Deluxe Video III* e titolazioni (*TV Text – TV Show*).

A questo punto si potrebbe pensare che mi serva un genlock... Ebbene, devo dirvi che ho avuto varie esperienze in proposito: ho iniziato comperandone uno di costo modesto (circa 500.000 lire) ma il segnale era alquanto instabile e sporco, sono passato poi a un altro genlock, vendendo il primo e acquistando il secondo per un valore pari a L. 1.500.000. Il segnale era nettamente migliore ma mancava sempre il sincronismo ed accendendo una o l'altra apparecchiatura mi saltava il quadro video; rivendendo allora il secondo, sono passato all'acquisto di un genlock da L. 2.500.000 e tutto andava bene, segnale e sincronismo. Mancava però una cosa importante e cioè la versatilità; ho quindi venduto anche il terzo (con il ricavato ho già acquistato appunto il secondo videoregistratore) e mi sono convinto che un bel mixer video, con tanto di fader e positioner è insostituibile e per questo entro fine anno penso di acquistare un bel Panasonic WJ-AVE5, oppure, chissà, un Panasonic WJ-MX10!

Ora, vi chiedo: esiste una scheda da usare al posto del modulatore, che abbia però un buon segnale videocomposito in uscita? Esistono in commercio delle apparecchiature che possano trasformare un segnale RGB in videocomposito? È neccessario montare il nuovo chip da poco commercializzato, il quale riesce a vedere un intero mega di memoria, per ottenere migliori risultati e minori problemi?

Edy Pellizzari
Ponte N. Alpi

Al posto del modulatore può utilizzare un encoder video (che trasforma il segnale da RGB a videocomposito PAL. La Telav (tel. 02/4455741-2-3) ne distribuisce più modelli. L'acquisto del nuovo chip è consigliato.

SUPER-RAM 1.5 AMIGA

Espansione a 2 MB per A-500, si inserisce nello slot sotto la tastiera al posto della vecchia espansione da 512 KB, completa di clock in tempo reale e batteria tampone.

Eccezionale! solo 269.000

ACCESSORI AMIGA

Articolo	Prezzo
Espansione 512 KB A-500	**99.000**
Espansione 512 KB con clock A-500	129.000
Hard disk A-590	**799.000**
Espansione 2 MB per A-590	**150.000**
Videon II'	450.000
Videon III'	550.000
DigiView 4.0	390.000
PAL-RGB converter	220.000
Drive esterno con switch	**165.000**
Drive esterno HD 1.44 MB Amiga	**259.000**
MiniGen A-500	299.000
Motherboard 2 slot 86 pin A-500	49.000
Mouse di ricambio	69.000
Espansione 2 MB A-2000	599.000
Hard card A-2091 40 MB A-2000	**950.000**
Espansione 2 MB per A-2091	**150.000**
Scheda de-interlacer per A-2000	450.000
Chip di espansione per A-3000	19.000 cad.
Tavoletta grafica seriale Amiga o PC	499.000
Fatter Agnus 8372/A	179.000
Interfaccia MIDI professionale	49.000
Penna ottica professionale	89.000
Alimentatore di ricambio A-500	129.000

SUPEROFFERTE NATALIZIE
FLOPPERIA

Articolo	Prezzo
Amiga 500	750.000
Amiga 500 con 30 giochi, mouse, joystick	790.000
A-500 con espansione a 1 MB	830.000
A-500 con esp. 1 MB e drive esterno	980.000
Amiga 2000 con hard disk 20 MB	1.950.000
Amiga 2000 con hard disk e 2 drive	2.170.000
Monitor colori 1084-S stereo	480.000
Genlock 2300 A-2000+TvShow+TvText	350.000
Amiga 3000 25 MHz 40 MB	6.300.000
Amiga 3000 25 MHz 100 MB	7.050.000

Tutto con garanzia originale Commodore

Viale Monte Nero 15
(nuova show-room)
20135 Milano
Tel. (02) 55.18.04.84 r.a.
Fax (02) 55.18.81.05 (24 ore)

Negozio aperto al pubblico tutti i giorni dalle **10** alle **13** e dalle **15** alle **19**.
Vendita per corrispondenza.

Prezzi IVA compresa

ATonce

Il fantastico PC/AT Emulator per Amiga 500 espansi a sole Lit. 499.000 IVA compresa

DUE COMPUTER IN UNO! Questa eccezionale scheda comprende 3 chip ad altissima tecnologia: un chip *custom*, il normale microprocessore dell'Amiga, Motorola 68000, e la CPU Intel 80286 da 8 MHZ, per rendere il vostro A-500 compatibile al 100% con un PC/AT.

Lo speciale *custom gate array* include un Bios originale, uso della porta seriale, parallela e mouse Amiga dal lato PC, supporta il suono, le schede grafiche CGA ed Hercules, espansioni ed hard disk Amiga, e tutto questo mentre usate normalmente AmigaDos in multitasking!

La scheda ATonce, incredibilmente compatta, si inserisce internamente sopra la CPU 68000, con un montaggio **semplicissimo e senza saldature:** è sufficente svitare qualche vite, una operazione che richiede non più di 3 minuti, dettagliatamente descritta nel manuale **in italiano**. ATonce, quando non è attiva, è totalmente trasparente nell'uso normale del computer. Richiede un Amiga espanso ad almeno 1 MB; dal lato AT la CPU 286 vede 640 KB Ram, ed ogni espansione di memoria del vostro Amiga oltre 1 MB sarà vista come memoria Estesa o Espansa. Ad esempio, se avete una SuperRam 1.5 (cioè con 2 MB totali), avrete un 286 con 640 KB da Dos + 1 MB di Ram EMS.

Se avete un hard disk AmigaDos autoboot, è possibile caricare Ms-Dos direttamente dall'hard disk Amiga, ed usare uno spazio su disco fisso per un massimo di 768 MB. Il blitter di Amiga è pienamente utilizzato per velocizzare le emulazioni Hercules 738x380 e CGA 640x200 con 4 o 8 colori. ATonce usa il drive interno di Amiga come drive "A:" da 720 KB, supporta tutti i drive esterni da 3"1/2 o 5"1/4, usa il mouse Amiga come mouse PC, configura la porta seriale come COM1 o COM2, configura la porta parallela come LPT1, supporta tutte le peculiarità dell'AT come la Ram Cmos con batteria tampone e real-time clock (converte l'equivalente di Amiga nelle espansioni con clock).

Disponibile in esclusiva presso:

Flopperia Srl, Viale Monte Nero 15
20135 Milano
Tel. (02) 55180484 r.a.

Megabyte, Piazza Malvezzi 14
25015 Desenzano D.G. (BS)
Tel. (030) 9911767 r.a.

SOFTWARE GALLERY

UNA GUIDA PER ORIENTARSI NEL MONDO DEL SOFTWARE

SNOWSTRIKE

L'erba del vicino è sempre da distruggere

Computer: C-64/Amiga
Supporto: Disco
Prezzo: L. 25.000/29.000
Produzione: Epyx/U.S. Gold
Distribuzione: Leader (Via Mazzini 1 21020 Casciago – 0332/212255)

Con perestrojka e glasnost ormai date per scontate e la Germania unita, i progettisti di videogiochi si sono trovati di fronte a un'impasse. Fino a qualche anno fa era facile stendere la trama di un gioco militare: bastava mandare i ragazzi americani alti, belli, biondi e dal bicipite possente a "fare giustizia" cancellando dalla faccia della terra qualche migliaio di sovietici brutti, neri, maleducati e male armati. La maggior parte dei grandi successi della Microprose si basa proprio su scenari di questo genere, ma la situazione attuale le rende anacronistiche simili ambientazioni.

Volendo fare le cose semplici, gli ideatori di questo programma avrebbero potuto mandare il loro supercaccia a distruggere le installazioni di qualche paese medio-orientale, ma c'era una possibilità ancora più interessante: il bersaglio di *Snowstrike* sono infatti i produttori di droga sudamericani, che in questo periodo di lotta particolarmente serrata agli stupefacenti si può dire sostituiscano più che egregiamente quella che era la "minaccia" rossa.

Bisogna purtroppo far notare subito che questo cambiamento di bersaglio è l'unica vera novità di *Snowstrike*, che risulta altresì un comune gioco di combattimenti aerei del tutto simile alla vasta concorrenza.

Il gioco si articola in missioni sul mare o sulla terra con bersagli come le raffinerie colombiane o i mezzi militari che le proteggono. Il pilota viene assistito dalla solita massa di strumenti digitali, minimamente coadiuvati questa volta da un secondo pilota che fornisce commenti più o meno sarcastici in un'apposita finestra del cruscotto.

Altra piccola variazione che, per quanto spettacolare, non influisce in realtà molto sul gioco vero e proprio è la possibilità di eiettarsi dal proprio caccia nelle situazioni critiche e controllare poi la propria discesa col paracadute.

Nella versione per Commodore 64 nonostante il programma non presenti alcuna pecca particolare, l'impressione che fornisce è di una mediocrità raccapricciante. La velocità di aggiornamento dello schermo, la risposta dei comandi, la (scarsa) complessità della grafica e la qualità degli effetti sonori sono tutti nella media per i programmi di questo genere. Anche la dotazione di armi e strumenti del nostro caccia è quanto di più standardizzato si possa immaginare, sembra persino di essere tornati a giocare con *F-15 Strike Eagle* della Microprose, prodotto già cinque anni fa sono.

Le schermate che appaiono fra una missione e l'altra e il sistema di

SCHEDA CRITICA

INSUFFICIENTE
Un pessimo prodotto che non merita nessuna considerazione.

MEDIOCRE
Il programma ha alcuni difetti di fondo, anche se nel complesso raggiunge quasi la sufficienza.

DISCRETO
Un prodotto accettabile, ma non aspettatevi grandissime emozioni

BUONO
Raccomandato vivamente: tra i migliori programmi della sua categoria.

OTTIMO
Eccezionale! Fino ad oggi non si era mai visto niente del genere

promozioni ricalcano pesantemente i giochi Microprose, tanto che alla fine viene da chiedersi se *Snowstrike* non fosse stato realizzato per questa software house e poi rifiutato per la sua mediocrità.

La versione per Amiga di *Snowstrike* si differenzia da quella per C-64 unicamente per la rapidità dell'aggiornamento dello schermo, che lascia piacevolmente stupefatti. La velocità di risposta dei comandi e quella degli oggetti sullo schermo è di gran lunga superiore a quella riscontrata in ogni altro metasimulatore per Amiga da noi provato, e il risultato è senz'altro più simile a una reale esperienza di volo supersonico. Da notare anche che in questa versione è possibile usufruire di un sistema di atterraggio automatico che manca sul C-64.

Siamo in vista dell'aeroporto nelle versioni per Amiga (sopra) e per C-64 (sotto)

Ci troviamo dinanzi a un prodotto che, senza infamia e senza lode, si piazza saldamente nella mediocrità più assoluta. Gli appassionati di simulazione di volo possono sicuramente trovare programmi più interessanti sui cataloghi Microprose, mentre chi cerca semplicemente un gioco riguardante i combattimenti aerei moderni può trovarsi a suo agio con *Snowstrike*... a meno che non possegga già un gioco equivalente.

F.R.

THE FINAL BATTLE

Il piacere dell'avventura anche per i videoguerrieri più navigati

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: PSS
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Vi ricordate di *Legend of the Sword*? Si trattava di un'avventura grafico-testuale controllata da icone che ci aveva fatto gridare al miracolo sul numero 5/88 di *Commodore Gazette* e che meriterebbe lo stesso voto anche oggi, a più di un anno di distanza.

The Final Battle altro non è che la seconda parte di *Legend of the Sword*, ma con una mossa veramente inaspettata; alla PSS hanno pensato bene di modificare completamente la struttura del programma. Il risultato è un'avventura grafica in isometria in cui il testo costituisce una parte minima rispetto a quanto rappresentato dalle immagini che si succedono sullo schermo.

Scopo del gioco è, nella migliore tradizione della letteratura fantasy, ritrovare una serie di oggetti magici che permettono di accedere alla fortezza del malvagio mago Suzar, sfuggito alla prigione arcana nella quale era stato imprigionato alla fine di *Legend of the Sword*. Del gruppo originario di cinque mercenari presentati nel primo gioco della serie ne sono sopravvissuti solamente tre, che all'inizio del gioco si trovano imprigionati in celle individuali in una misteriosa prigione.

Il giocatore dovrà riuscire a scappare dalla propria cella, recuperare i compagni e partire alla ricerca di sette gemme fatate. Una volta recuperati questi oggetti bisognerà penetrare nel dominio di Suzar, dove si avranno solo poche ore a disposizione per recuperare le uniche due armi capaci di eliminare il negromante e, naturalmente, riuscire a sconfiggerlo.

The Final Battle è controllato ancora una volta interamente da icone, che questa volta circondano l'area di gioco. Benché il loro utilizzo non sia dei più immediati è sufficiente un minimo di pratica per riuscire a realizzare rapidamente ogni azione desiderata, raggiungendo a volte un notevole livello di manipolazione degli elementi offerti dal gioco.

Come nel programma precedente anche qui la grafica viene accompagnata da un certa quantità di testo di buona qualità, tuttavia in quantità molto minori rispetto a *Legend of the Sword*.

Se la trama dell'avventura merita un plauso grazie alla sua struttura mai

banale e piena di tocchi veramente intelligenti, ciò che colpisce di più in *The Final Battle* è senza dubbio la qualità e la varietà della sua grafica, che raggiunge spesso livelli di dettaglio davvero eccezionali. Nonostante la mappa di gioco sia decisamente vasta (tanto che è stata inclusa un'utilissima funzione di automapping), le ripetizioni grafiche sono poche e perdonabili, e tutto ciò va a contribuire al realismo dell'atmosfera creata.

È impossibile dire altro riguardo a questo gioco senza svelarne problemi e soluzioni, se non che la necessità di far collaborare i vari personaggi lo pone sullo stesso piano dei giochi di ruolo più semplici. Questa caratteristica non mancherà d'interessare chi si è ormai stufato dei numerosissimi incontri tuttavia mancanti di spessore della maggior parte degli RPG veri e propri e cerca un tipo di gioco più raffinato e dettagliato.

Non possiamo concludere questa recensione che con un invito a provare *The Final Battle*: un programma veramente ottimo che restituirà il piacere dell'avventura anche ai videoguerrieri più navigati. **F.R.**

PLOTTING

L'ultimo discendente di Tetris

Computer: Amiga
Supporto: Disco
Prezzo: L. 29.000
Produzione: Ocean
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Abbiamo già avuto modo di notare in passato come il successo ottenuto da *Tetris* abbia dato inizio a una vera e propria invasione di giochi più o meno intelligenti basati su poligoni, forme colorate ed elementi grafici di ogni sorta. *Plotting* è uno degli ultimi prodotti di questa immensa famiglia, e riesce a distinguersi dai suoi molti parenti sotto diversi aspetti.

Lo scopo del gioco è di fare scomparire un certo numero di elementi da un agglomerato di blocchi quadrati disposto sullo schermo. Per farlo, il giocatore controlla una strana creatura amorfa capace di muoversi sul lato dello schermo opposto ai cubi. Il buffo personaggino comincia la partita "impugnando" un cubetto sul quale è impresso uno dei cinque simboli che contraddistinguono anche quelli accatastati poco distante. Lanciando il blocco contro un cubetto decorato con lo stesso simbolo entrambi scompariranno, provocando nel contempo l'espulsione di un nuovo blocco dalla catasta. Quest'ultimo elemento rimbalza a fianco dell'ameboide, che può così procedere nella sua opera di distruzione.

Sebbene a parole *Plotting* possa sembrare decisamente poco intelligente, giocarlo è tutto un altro paio di maniche. Sin dal secondo livello bisogna pianificare con una certa attenzione le mosse da compiere: in ogni partita ci si può permettere di rimanere in situazioni di stallo (senza la possibilità di colpire alcun blocchetto del giusto tipo) solo due volte, e la presenza di un limite di tempo per finire il quadro rende il tutto più difficile.

Sebbene il gioco sia già abbastanza divertente così come l'abbiamo descritto, il programma possiede anche altre caratteristiche degne d'interesse. Tanto per cominciare è possibile giocare in coppia – ognuno nella propria finestra di schermo – con un secondo giocatore: riuscendo a far sparire più di un blocchetto per volta

tutti i pezzi espulsi dalla propria formazione andranno a materializzarsi nella finestra avversaria, complicando così le cose.

Una seconda possibilità è quella di personalizzare il gioco progettando i propri livelli con l'editor incorporato, benché i ragionamenti richiesti per strutturare un nuovo schermo siano ben più complicati di quelli coinvolti nella sua soluzione.

Con la sua grafica coloratissima e la musichetta allegra, *Plotting* è un'ottima conversione di un gioco a gettoni relativamente sconosciuto, che dovrebbe costituire una piacevole variante per chi è abituato ai (generalmente) più complessi giochi Amiga. Per un gioco di questo genere la realizzazione tecnica non può essere niente di particolarmente eccelso, tuttavia *Plotting* possiede il fascino comune a tutti i puzzle game d'azione, e dovrebbe rimanere interessante anche a tempi lunghi.

F.R.

CADAVER

I Bitmap Brothers vanno all'avventura

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: Bitmap Brothers, Image Works
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Non si può certo dire che i Bitmap Brothers manchino di fantasia: dopo aver stupito mezzo mondo con l'azione frenetica di *Xenon II* poco meno di un anno fa, il loro prodotto successivo è qualcosa di completamente diverso dall'incredibile shoot'em up ambientato nello spazio.

Cadaver è infatti un'avventura. Un'avventura né più né meno come quelle cui ci aveva abituato a suo tempo la grande Infocom, ma con un'importante differenza. Mentre nei giochi della Infocom era solo il testo a rievocare ambientazioni fantastiche e azioni rocambolesche, in *Cadaver* tutto il gioco si basa sulla grafica. Presentato con una prospettiva isometrica, è ambientato in un castello semiabbandonato che sorge nel bel mezzo di una fetida palude. Il personaggio che controlliamo, il nano Karadoc, è stato incaricato di sopprimere il negromante che ancora vi abita in cambio di tutti i tesori che riuscirà a recuperare durante l'esplorazione del maniero.

Karadoc si può muovere in ogni direzione, saltare e interagire con

l'ambiente e gli oggetti circostanti grazie a una serie di menu a icone che appaiono ogni volta che ci si avvicina a qualche elemento d'interesse. I menu si differenziano automaticamente a seconda della natura dell'oggetto, per cui le scelte e l'azione risultano sempre piuttosto rapide e – una volta impadronitisi del sistema – di facile esecuzione.

Lo svolgimento del gioco è, a parole, piuttosto banale: si esplora un tetro ambiente sconosciuto, si combattono mostri e si recuperano tesori. Si cerca di sfruttare al meglio un criptico sistema di magie e alla fine si elimina il nemico più potente di tutti. Una simile descrizione deprimerebbe anche il più incallito ottimista, ma una volta caricato il gioco ogni giudizio cambia. La realizzazione grafica degli ambienti e dei personaggi che li popolano è stupenda, e le animazioni si sprecano. I menu personalizzati permettono un livello di manipolazione degli oggetti notevolissimo, e benché non siano possibili azioni complicatissime come quelle permesse dalle avventure testuali della Magnetic Scrolls, quel che viene offerto è più che sufficiente.

Il livello di sofisticazione del gioco è altissimo: in alcuni ambienti si può sperimentare l'effetto delle maree, alcuni problemi si basano su macchinari tanto complessi da fare invidia a Wile E. Coyote, la varietà dei mostri garantisce incontri sempre nuovi a ogni svolta. Raggiungere la fine del gioco è un'impresa che impegnerà molto a lungo anche il videoavventuriero più smaliziato, tanto che nella confezione è contenuto un documento da spedire alla Mirrorsoft nel caso si riuscisse a sconfiggere il negromante, che a sua volta garantisce l'invio di un attestato di merito firmato dagli stessi autori.

Il sonoro è stato usato con parsimonia ma molto oculatamente: brontolii in lontananza e rumori naturali assortiti non fanno che contribuire all'atmosfera del gioco e aumentare il livello di tensione che si accompagna a ogni esplorazione di un ambiente tanto inquietante.

Poiché il gioco richiede senza dubbio parecchie partite per essere completato, è possibile salvare la partita in corso per riprenderla in seguito. Fin qui non ci sarebbe niente di strano: la novità sta nel fatto che più si procede verso la fine del gioco e più bisognerà "pagare" (con i tesori raccolti) per portare a termine questa operazione. Meno male che i caricamenti sono gratis!

Cadaver è un programma di altissima qualità che prosegue ottimamente l'onorata tradizione dei Bitmap Brothers: sicuramente il miglior prodotto per gli appassionati delle avventure fra i molti titoli di questo genere usciti in questo periodo, dovrebbe essere in grado di interessare tanto gli avventurieri quanto gli amanti dell'azione.

F.R.

MURDER

Una versione deludente del gioco da tavolo Cluedo

Computer: C-64/Amiga
Supporto: Disco
Prezzo: L. 25.000/29.000
Produzione: U.S. Gold
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Sembra che la moda di questo mese nel campo dei videogiochi sia costituita dalle avventure dinamiche in isometria. Si tratta di quel genere di programmi che rese famoso a suo tempo un computer pieno di difetti come lo ZX Spectrum, grazie alla sua resa spettacolare. Questo mese abbiamo avuto infatti occasione di provare, oltre a questo *Murder*, anche *Cadaver* e *The Final Battle* della Mirrorsoft, tutti titoli che sono basati su questa tecnica di rappresentazione grafica.

Murder è probabilmente il titolo meno interessante dei tre e risulta decisamente limitato anche per via del difficile confronto diretto che è stato imposto dalla concorrenza. È poco più che una versione su video del famosissimo gioco da tavolo *Cluedo*, in cui il giocatore entra nei panni di un investigatore dilettante impegnato a trovare la soluzione di un omicidio.

Il gioco è ambientato all'interno di una tipica residenza di campagna inglese, e la principale difficoltà è costituita da un limite di tempo di due ore reali, passate le quali arriverà sul luogo del delitto una squadra di poliziotti "veri" provenienti da Londra e la partita sarà da considerarsi persa.

La principale caratteristica di questo programma (sbandierata davvero in ogni angolino della confezione e dei comunicati stampa) risiede nel numero apparentemente "infinito" di casi sui quali si può investigare... Ma naturalmente la realtà è ben più deludente di quanto annuncia la pubblicità.

Proprio come nel già citato *Cluedo*, ogni caso è composto di una serie di elementi che differenziano minimamente un omicidio dagli altri: il locale in cui si è consumato il fatto crimino-

so, l'arma impiegata, la vittima e, naturalmente, il colpevole. Mentre per stabilire la vittima e l'arma del delitto è sufficiente girovagare per il maniero esaminando da vicino ogni anfratto, identificare il criminale richiede una procedura leggermente più complessa.

In una prima fase bisognerà interrogare tutti i presenti per stabilire dove questi si trovassero al momento dell'omicidio, cosa stessero facendo e per valutare l'affidabilità dei loro alibi.

Una volta tentata una prima ipotesi, la maggiore difficoltà sta nel trovare il modo d'impadronirsi delle impronte digitali del sospettato per poterle confrontare con quelle presenti sull'arma del delitto. Per farlo è sufficiente avvicinare al personaggio in questione un oggetto adeguatamente ripulito, che prima o poi lo impugnerà marchiandolo quasi indelebilmente. A questo punto si clicca sull'icona preposta al confronto delle impronte e... basta.

Murder finisce così, lasciando il giocatore con un sapore di deja vu e di "si poteva fare di meglio". La prima partita con un programma simile può anche essere divertente, ma senza la competitività che con-

traddistingue il *Cluedo* originale, generata dalla presenza di più partecipanti ogni altro caso diventa una noiosa ripetizione di una routine già di per sé piuttosto poco entusiasmante.

Si tratta di un prodotto consigliabile davvero solo ai lettori più instancabili dei Gialli Mondadori. *Murder* soffre di una monotonia davvero mortale; peccato, perché l'interessante stile grafico (tutto in bianco e nero, proprio come un vecchio film poliziesco) avrebbe potuto essere sfruttato molto, molto meglio.

F.R.

WINGS

Volare senza strumenti: ritornano aerei più a misura d'uomo

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: Cinemaware/Mirrorsoft
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Nel corso degli anni gli utenti di computer hanno imparato a pilotare i velivoli più sofisticati, dalle cannoniere elitrasportate ai caccia invisibili, dagli alianti acrobatici alle astronavi intergalattiche. Qualunque possessore di Amiga è ormai in grado di far volare anche i mezzi più avanzati, e sta diventando sempre più difficile trovare interessanti i combattimenti a mach 2 offerti dalla maggior parte dei giochi di volo.

Alla Cinemaware qualcuno si dev'essere accorto di questa situazione e deve avere deciso di "riportare la magia" in questo genere di programmi. La soluzione, estremamente originale, è stata una volta tanto di fermarsi a guardare il passato.

Wings è ambientato proprio agli inizi della storia del volo, tanto che si apre addirittura con un "documentario" riguardante il primo decollo a opera dei fratelli Wright. Il gioco vero e proprio si svolge all'incirca un decennio dopo quell'importantissimo evento, durante la Prima guerra mondiale.

Il giocatore impersona un impavido pilota britannico di biplani, alle prese tanto con i nemici tedeschi quanto con le nuove e poco affidabili tecnologie di cui è pioniere.

I combattimenti aerei nell'epoca del Barone Rosso sono molto diversi da quelli cui siamo abituati. Innanzitutto, i velivoli in dotazione si spostano a velocità molto basse, permettendo anche complesse manovre in spazi ridottissimi. In secondo luogo le armi in dotazione hanno un raggio piuttosto limitato, e non si possiede assolutamente alcuno strumento che possa aiutare nel volo!

Dopo anni di postbruciatori e sistemi di contromisure elettroniche trovarsi a bordo di un trabiccolo che stalla a ogni piè sospinto con solamente un piccolo mirino fisso davanti agli occhi è veramente traumatico. Fortunatamente, i progettisti del gioco hanno pensato di dare una mano all'utente, sfruttando la testa del pilota raffigurato sullo schermo come una sorta di radar primordiale, facendola ruotare nella direzione dell'avversario più vicino.

Gli scontri in 3D con i mezzi germanici rappresentano la parte più cospicua del gioco, che ripercorre le fasi finali della Prima guerra mondiale. Qui si vola affiancati da alcuni membri della propria squadriglia, con i quali bisogna cercare di operare al meglio nonostante la mancanza di

qualsiasi sistema di comunicazione tra i piloti: ovviamente non c'è infatti nessuna possibilità di usufruire di una radio

Wings non si limita tuttavia a una semplice serie di duelli fra biplani e triplani. Con il progredire del gioco bisognerà infatti dedicarsi anche a missioni di bombardamento e di attacco al suolo raffigurate bidimensionalmente, ma non per questo meno divertenti.

A completare il programma si aggiunge poi tutta una serie di schermate di testo, intermezzi animati e sequenze musicali che riescono a ricreare piuttosto bene l'atmosfera di inizio secolo. Quando il proprio pilota viene ferito o abbattuto in missione si arriva persino a sentire la mancanza di un vecchio amico (e sì, i giochi cominciano a darci alla testa).

Come in ogni prodotto Cinemaware la realizzazione tecnica del programma è ineccepibile. La sequenza tridimensionale è estremamente realistica, e sono state tenuti in considerazione elementi minuti quali l'influenza del senso di rotazione dell'elica sulla traiettoria di volo del nostro aereo.

Le sezioni in 2D non sono in realtà proprio niente di particolarmente eccezionale o innovativo. In ogni caso, si può comunque dire che si amalgamano bene con le altre sequenze e spezzano piacevolmente il ritmo della vicenda.

È inoltre molto importante sottolineare che le animazioni sono molto fluide. La mancanza di eccessiva raffinatezza di strumentazione va infatti a beneficio di un velivolo relativamente facile da controllare e dalla docile risposta ai comandi.

Wings è in definitiva un ottimo programma, reso ancor più interessante dalla novità costituita dal tipo di veicoli manovrati. Tutti gli appassionati di simulazioni di volo e di giochi di guerra dovrebbero provarlo: dopo una partita a *Wings* si arriva immancabilmente a benedire gli inventori dell'altimetro, del radar e dei missili a ricerca di calore! Nello stesso tempo si apprezza però anche quel modo di volare "più umano" degli inizi del nostro secolo.

F.R.

NIGHTBREED

Due passi nell'horror con Clive Barker

Computer: Amiga
Supporto: Disco
Prezzo: L. 29.000
Produzione: Ocean
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Dopo il grande successo internazionale di *Indiana Jones – The Graphic Adventure* della Lucasfilm, la Ocean deve aver pensato bene di produrre qualcosa di simile sfruttando il più possibile qualche altro film. La scelta è caduta evidentemente su questo *Nightbreed* (distribuito in Italia con il titolo *Cabal*), basato su un romanzo del famoso autore horror Clive Barker. Dalla software house inglese sono usciti – proprio come per *Indiana Jones* sia un gioco d'azione sia la "versione interattiva" del film che abbiamo avuto modo di esaminare.

Bisogna subito dire che purtroppo *Nightbreed – The Interactive Movie* ha ben poco del grande fascino che sprigionava il titolo Lucasfilm. Piuttosto che di una singola avventura articolata coerentemente, qui troviamo una serie di giochi d'azione di media qualità riuniti da un paio di schermate di grafica e testo. Il risultato finale non è niente di orribile, tuttavia il sottotitolo appare sotto questa luce un po' troppo pretenzioso.

Il giocatore interpreta il ruolo di Boone, un uomo psichicamente instabile che viene ingiustamente accusato di aver compiuto una serie di efferati omicidi. Ricoverato in un ospedale psichiatrico per subire le cure del caso, Boone scopre dell'esistenza di una città sotterranea chiamata Midian nella quale vive una razza di creature mostruose ma tendenzialmente pacifiche, che hanno scelto la reclusione piuttosto che gli scontri quotidiani con l'intollerante razza umana.

Fuggito dall'ospedale, Boone scopre l'ingresso di Midian. Il problema sarà quello di farsi accettare dalla sua strana popolazione, anche perché il buon Boone è stato seguito da un misterioso personaggio chiamato "la Maschera", il vero responsabile degli omicidi. Sulle tracce di entrambi è poi la polizia, i cui membri pensano bene di demolire Midian e le sue abominazioni in un eccesso di xenofobia.

Il gioco ripercorre tutte le tappe di questa vicenda: per riuscire a progredire la cosa più complicata sta nel trovare la sequenza esatta in cui affrontare i sottogiochi, che risultano letali se giocati disordinatamente.

Fra i vari giochi ci sono sequenze di labirinto (gli spostamenti in auto su una mappa costellata di posti di blocco), di pura coordinazione occhio-mano (una fuga tridimensionale dalle "attenzioni" di una belva cannibale), di platform game (quelle all'interno di Midian) e di tiro al bersaglio, ma ben poca di quella logica e strategia che ci si aspetterebbe di trovare in una presunta avventura dinamica.

La qualità di ogni sezione è discreta, senza grosse disattenzioni o grandi innovazioni tecniche, e dovrebbe soddisfare tutti gli amanti dei giochi d'azione con un certo gusto per l'horror. La grafica, pur non particolarmente curata, è all'altezza del suo compito, così come il sonoro campionato, a volte realmente agghiacciante.

Ho avuto occasione di parlare personalmente con Clive Barker riguardo la realizzazione di questo gioco, alla quale lo scrittore ha partecipato ampiamente in termini di progettazione. Durante l'incontro Barker si è detto "affascinato dall'alta tecnologia ma ancora un po' perplesso della magia che permette di muovere mostri sullo schermo dei computer". Purtroppo nel risultato finale queste perplessità sono tutte ben visibili: forse sarebbe stato meglio lasciar progettare il gioco a un professionista, che avrebbe potuto dare un maggior spessore a una serie di giochini altrimenti piuttosto comuni.

F.R.

DRAGON STRIKE

Un simulatore di volo estremamente originale per la serie di AD&D

Computer: Amiga
Supporto: Disco
Prezzo: L. 69.000
Produzione: TSR/SSI
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

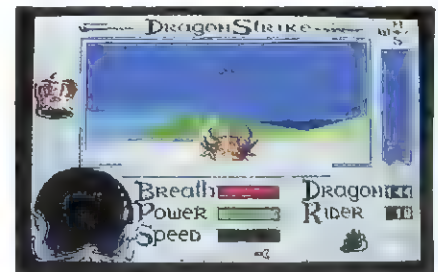
Come accade ormai da diversi numeri, *Commodore Gazette* continua la sua rassegna dei prodotti ispirati al gioco di ruolo *Advanced Dungeons & Dragons*. Già parlando di *War of the Lance* (*Commodore Gazette* numero 3/90) abbiamo avuto occasione di descrivere l'importanza del ciclo di *Dragonlance* nel mondo dei giochi di ruolo. Sino a oggi questa branca di *Advanced Dungeons & Dragons* sembrava essere stata trasposta in ogni formato possibile e immaginabile su home computer: giochi di ruolo, war game, avventure dinamiche...

Ma l'immaginazione dei progettisti non ha davvero mai fine, e *Dragon Strike* dimostra come si possa tirar fuori un prodotto interessante anche da un genere che apparentemente è già sovrasfruttato da ogni tipo di opera digitale.

Si può infatti dire con tranquillità che il programma del quale ci occupiamo questo mese non appartiene ad alcuna categoria conosciuta: come si può descrivere altrimenti un simulatore di drago?

Il gioco riprende con una fedeltà piuttosto discutibile gli eventi militari descritti negli ultimi libri della seconda trilogia di *Dragonlance*, mettendo l'utente nei panni di uno dei campioni impegnati nella difesa del Bene dall'avanzata della Regina Nera Takhisis.

Non ci sarebbe proprio nulla di strano se le battaglie si combattessero con i mezzi ai quali siamo tutti abituati... Ma in uno scenario fantasy come quello offerto dal singolare mondo di Krynn, le forze in campo non si spostano con niente di meno imponente di draghi o intere città volanti!

La schermata principale del gioco ci vede saldamente assicurati al dorso di un drago: sotto l'ascella sinistra il nostro personaggio stringe una Dragonlance – unica arma in grado di danneggiare questi pachidermi volanti – mentre si suppone che nella mano destra impugni le corregge di cuoio necessarie a dirigere lo strano destriero.

Scopo del gioco è di portare a termine una serie di missioni assegnate dal comando dell'Ordine dei Cavalieri Solamnici.

Tornando alla base ancora in vita, alla fine di una missione completata con successo, si ottengono onorificenze, migliori cavalcature e, naturalmente, nuove missioni sempre più complesse e rischiose (forse non vale la pena di tornare...).

Se il programma si limitasse a far

svolazzare il giocatore sul paesaggio a poligoni pieni di Krynn, l'interesse per il gioco morirebbe subito. Fortunatamente, i progettisti hanno invece sfruttato pienamente le caratteristiche della cavalcatura, che possiamo così utilizzare come e meglio dei sofisticati armamenti a bordo dei caccia futuribili cui siamo senz'altro più abituati. Oltre alla Lancia abbiamo infatti a disposizione diversi tipi di simpatici soffi letali (diversi a seconda del drago che viene cavalcato), gli artigli resistentissimi dell'animale e la sua coda, che può essere usata come arma una volta che si ha acquisito una certa padronanza della bestia.

Il gioco favorisce per sua stessa natura le battaglie molto ravvicinate: gran parte del divertimento sta proprio nell'imparare a manovrare in modo da toccare i propri avversari, stringerli fra gli artigli e lottare – magari mentre si sta cadendo in vite verso terra – fra le spettacolari traiettorie degli altri draghi coinvolti nella lotta.

A disposizione del giocatore c'è una serie di strumenti "magici" che lo aiutano in battaglia. La loro funzione è essenzialmente quella dei molto più prosaici altimetri, radar e altri strumenti imbarcati sui jet, e imparare a leggere fra i riflessi di una sfera di cristallo si rivela una abilità vitale.

Dragon Strike su Amiga funziona a una velocità notevole, decisamente maggiore di quella che si può ottenere con lo stesso gioco sui PC più accelerati. L'azione scorre fluida e piacevole, anche se gli elementi del paesaggio e le cittadelle volanti (realizzati in grafica poligonale piena) sono lungi dall'essere realistici. La meccanica degli scontri fra draghi può raggiungere livelli di complessità piuttosto alti, e riesce a impegnare anche i giocatori più smaliziati.

La resa grafica è buona, quella del sonoro è invece forse un po' al di sotto della media, ma comunque accettabile. La confezione contiene inoltre una serie di bellissime schede riassuntive delle caratteristiche di ogni drago, una vera chicca per tutti gli appassionati del ciclo di *Dragonlance*.

Per ogni amante del genere fantasy è un gioco da non perdere.

F.R.

KICK OFF 2

La fretta da mondiali ha sortito effetti negativi

Computer: C-64/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 19.500/25.000/29.000
Produzione: Anco Software
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Credo che non ci sia possessore di Amiga (e non solo di Amiga) che non conosca, almeno per "sentito dire" l'ormai mitico *Kick Off*. Nel corso degli ultimi due anni, la famiglia si è via via arricchita, prima con una sorta di "appendice", riservata a chi già possedeva *Kick Off* (l'eccezionale *Extra Time*), poi con *Player Manager*, che ha introdotto l'aspetto "gestionale" di una squadra di calcio, e con *World Cup Kick Off*, dedicato ai mondiali.

Ed eccoci finalmente alla tanto attesa uscita del seguito: *Kick Off 2*! Molte le nuove caratteristiche promesse: calci di punizione con barriera, fuorigioco, rovesciate, possibilità di disegnare la propria divisa di gioco e tantissime altre ancora. Eppure, il voto non eccezionale lascia presagire che qualcosa è andato storto. Forse le promesse non sono state mantenute? In parte è così, ma procediamo con ordine. Innanzitutto, sembra che sia stata la troppa fretta di fare uscire il gioco sul mercato, per poter sfruttare l'ondata trascinante della febbre "mondiale", a far sì che troppe caratteristiche diano l'impressione di essere state inserite alla bell'e meglio, tanto per "far numero" e poter essere inserite nella pubblicità sul retro della scatola. Per fare un esempio, la già orribile traduzione in italiano si rivela peggiore che mai: i famosi "cartellino gaillo" e "fine de la partita" sono sempre presenti, altri termini sono anche peggiorati, come "calcio d''angalo" con due apostrofi, e a loro se ne è aggiunta un'intera e nutrita serie di nuovi.

Dopo l'interminabile caricamento ci troviamo nel menu principale, che presenta, oltre a quelle già conosciute, molte nuove opzioni. Allenamento consente di fare pratica nei tiri e nei passaggi e non è mutato rispetto alle versioni precedenti. Eventi Speciali permette di caricare eventuali dischi dati che potrebbero essere pubblicati in futuro. Sul disco stesso di *Kick Off 2* viene gratuitamente (si fa per dire) fornito il primo di questi "eventi": si tratta, guardate un po', proprio della coppa del mondo '90.

Replay rappresenta invece una delle novità veramente piacevoli di *Kick Off 2*. È possibile infatti in qualunque momento del gioco rivedere gli ultimi secondi giocati, e non solo dopo aver segnato una rete, a velocità normale o dimezzata ed eventualmente salvare su disco fino a due (e non come sostiene il manuale tre) replay per ogni partita. Se si desidera sfruttare questa possibilità è necessario scegliere il nome del file in cui salvare questi dati prima d'iniziare la partita, tramite questa opzione del menu principale, pena la sovrascritura di file omonimi precedenti. Sono presenti anche alcune semplici funzioni di "montaggio" del replay.

Altra grande novità è la possibilità di disegnare la propria maglia. È possibile scegliere tra sei diversi stili di base (sempre a differenza di quanto dice il manuale, tutti gli stili sono disponibili e non solo i primi due) da colorare a piacere (o quasi).

La voce Opzioni porta alla comparsa di un nutritissimo menu contenente tutte le scelte necessarie a personalizzare il gioco secondo i propri gusti e necessità. Si può scegliere su che tipo di terreno giocare, se sul normale campo erboso, su un campo bagnato, fangoso o artificiale. A ogni campo corrisponde una diversa rappresentazione grafica, anche se, a dire la verità, esteticamente non sono proprio il massimo. La durata della partita può assumere quattro diversi valori: 2 x 3 minuti (come in *Player Manager*), 2 x 5 (per default), 2 x 10 e 2 x 20 minuti. Anche il vento può essere regolato a piacere: da senza, a leggero, a medio e infine a forte. Gradevole novità il fatto di poter decidere se andare ai supplementari (di durata proporzionale alla partita), ed eventualmente ai rigori, in caso di parità al termine dei tempi regolamentari. Il doppio effetto o aftertouch è un'altra caratteristica vista per la prima volta in *Extra Time* che non poteva mancare. Consente infatti, se attivata, di deviare la palla per qual-

che frazione di secondo dopo il tiro e di ottenere quindi tiri a effetto in grado di "spiazzare" completamente anche il miglior portiere. È possibile, poi, scegliere il livello di abilità di ognuna delle due squadre fra le solite cinque "gradazioni", da C2 a Internazionale, oppure caricare una squadra creata con *Player Manager*.

Il menu Opzioni, infine, ci dà anche la possibilità di selezionare quattro tattiche di gioco da utilizzare nel corso della partita. Possono essere scelte fra le 8 già presenti sul disco oppure fra quelle eventualmente create su un disco dati, sempre nel caso si possieda *Player Manager*.

Per entrare nel gioco vero e proprio, vi sono ben quattro possibilità: la classica "partita secca", un'amichevole internazionale, un campionato fra otto squadre e una coppa a eliminazione diretta sempre fra otto squadre. È possibile scegliere se giocare da soli, in due, in tre o in quattro. Le due ultime possibilità sono riservate, però, ai possessori di un fantomatico adattatore per il collegamento di più joystick. Giocando in due ci si può schierare entrambi contro il computer, con la limitazione però di dover giocare un ruolo "fisso".

Una volta determinato il numero dei giocatori si passa alla schermata per la selezione dei ruoli, presa di peso da *Player Manager*, e qui cominciano le dolenti note. Il manuale cita con orgoglio il fatto che ogni giocatore possiede ben sette caratteristiche variabili, che influenzano il suo gioco, come velocità, resistenza, agonismo... È vero, lo abbiamo verificato con *Player Manager*, ma non vi è alcuna possibilità in *Kick Off 2* di sapere quale sia il valore di queste caratteristiche, se non cercando d'indovinarle con un'attenta osservazione del comportamento dei giocatori durante la partita. Senz'altro molto scomodo.

Grosse delusioni attendono anche chi si aspettava di poter creare la propria squadra del cuore con ogni giocatore nel proprio ruolo. Niente da fare: i nomi dei giocatori sono immutabili e assurdi: si va da un "S.D.R. De Agostini" che diventa poi "De Agosti" perché la stringa è troppo lunga, a "G.Bosco", carico di reminiscenze religiose. Il fatto, poi, che i cognomi disponibili siano necessariamente limitati, fa sì che intere famiglie militino nella stessa squadra.

Finalmente entrati nella partita, vediamo che la grafica, eccettuate le magliette dei giocatori, è quella di sempre. C'è qualche modifica presa dal solito *Player Manager*: il nome del giocatore in possesso di palla e il risultato sono infatti sovraimpressi al campo di gioco. Peccato, però, che la barra orizzontale gialla che nella versione precedente serviva a contenere queste informazioni, non sia stata sostituita da un allargamento dell'area di gioco visibile ma semplicemente dal nero di fondo. Cambia anche il pallone, non più di un bianco uniforme, ma cosparso di puntini neri la cui animazione non è quel che si dice entusiasmante.

Ma che cosa è mutato nel gioco in sé? Innanzitutto si ha ora un maggiore controllo nella rimessa laterale: diventa ancora più facile passare la palla direttamente agli avversari. Il calcio d'angolo è stato ridisegnato: la finestra per la scelta della direzione è speculare a quella cui eravamo abituati (perché? Boh...) ed è possibile dosare la forza del tiro a seconda della durata della pressione del pulsante di fuoco. Si va da circa mezzo metro all'orbita circumlunare. Difficile ottenere i valori intermedi.

Il nuovo calcio di punizione con barriera, che presenta una miriade di opzioni, mal riportate sul manuale, si rivela in pratica più coreografico che altro, essendo estremamente difficile ottenere un tiro che sia forte, preciso e imprendibile tutto nello stesso tempo. Un bug nel programma fa sì che, ogni tanto, il timer del gioco si fermi dopo uno di questi tiri, e non riparta che dopo un goal, un rigore, o un secondo calcio di punizione. L'annunciato fuorigioco non esiste proprio, e così pare anche la rovesciata "a bicicletta", che è però riportata dal manuale. Il portiere ha fatto invece dei progressi incredibili. Tanto era "tonto" quello di *Kick Off* originale e già piuttosto sveglio quello di *Extra Time*, così questi nuovi portieri sono delle vere e proprie "belve". Parano tiri incredibili, rubano la palla agli attaccanti prima che questi si siano resi conto di essere entrati in area, e non escono più a casaccio su calcio d'angolo. Costruire una buona azione nell'area avversaria diventa quasi impossibile, e spesso conviene tirare al volo e affidarsi alle meraviglie del doppio effetto.

Un simile miglioramento nel portiere lo si deve pagare in qualche modo, però, e lo si vede nel comportamento degli altri giocatori. Dal canto loro, infatti, oltre ad alcune piacevoli novità – la possibilità di infortuni e le sostituzioni – presentano due lati negativi. Il primo, che è semplicemente una questione di abitudine, viene dal fatto che i giocatori, a parità di tattica usata, non sono più al loro posto abituale. Il cross a centro area a cercare il numero 9 si risolve in genere nel tirare in un'area deserta, con i numeri 9 e 10 che arrancano in distanza. Mentre ciò si può risolvere mutando i propri schemi di gioco, non è così facile rimediare al secondo difetto: l'apparente "inebetimento" dei giocatori. Non è raro vedere un lancio che attraversa l'intera area di rigore, zeppa di attaccanti, senza che nessuno di loro si sogni minimamente di intervenire e senza che il programma ci permetta di controllarne qualcuno per tempo. Inoltre, i giocatori tendono spesso a fare delle sciocchezze madornali "di testa propria": falli assurdi, tackle invece di stop, stop invece di tiri, rimesse laterali direttamente sul fondo e tante altre variazioni che non migliorano certamente l'umore di chi gioca. Per quanto riguarda il controllo dei giocatori, il miglior titolo della serie rimane sempre *Extra Time*. A completamento del tutto, il disordinato e mal tradotto manuale, che si riferisce evidentemente a una versione non definitiva del gioco. Mancano molte cose, prima fra tutte la descrizione del tasto "F", che funziona come in *Player Manager* e permette di seguire uno qualsiasi dei 22 giocatori in luogo della palla (indispensabile per chi intende giocare in un singolo ruolo).

Kick Off 2 e i suoi parenti prossimi rimangono comunque infinitamente superiori a qualsiasi concorrente nel campo della simulazione calcistica. Il voto "discreto" assegnato a quest'ultima versione dipende in larga misura da quello che il programma avrebbe realmente potuto essere, e senza eccessive difficoltà, se solo durante la sua realizzazione gli fossero stati dedicati un po' più di tempo e di attenzione. Speriamo in un *Extra Time 2*.

A.P.T.

THEIR FINEST HOUR

La Battaglia d'Inghilterra rivissuta sull'Amiga

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: LucasFilm Games
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Il gioco consente di porsi ai comandi di ben otto aerei diversi, per poter rivivere ogni aspetto del conflitto. Due velivoli sono di nazionalità inglese: si tratta dei famosi caccia Spitfire e Hurricane, i più noti aerei utilizzati dalla RAF. Sono invece sei gli apparecchi dal lato germanico: il potente caccia Messerschmitt Bf 109E e il pesante ma pericoloso biposto Bf 110 "Zerstorer"; inoltre quattro bombardieri: il tristemente famoso Junkers Ju 87B "Stuka", il Dornier Do 17 "matita volante", l'Heinkel He 111 e lo Junkers Ju 88. Per ognuno di questi aerei è disponibile una vasta gamma di missioni, incluse quelle di "training", di difficoltà e complessità variabili, ma sempre ben bilanciate. È presente inoltre un'opzione per poter giocare l'intera Battaglia d'Inghilterra in modo che il risultato di ogni missione influisca sulle successive. Una vera "chicca" è rappresentata dal fatto di potersi poi creare a piacere delle nuove missioni grazie a un'apposita utility fornita con il programma.

Una volta scelto lo scenario, è possibile accedere a una schermata di testo contenente la sua descrizione, completa di note storiche e piano di volo. Si possono modificare parametri quali la quantità di munizioni a disposizione, la resistenza ai danni dell'aereo e l'abilità dei piloti nemici. Ulteriore profondità al gioco viene data dalla possibilità di crearsi un proprio disco dati sul quale salvare i piloti e le loro statistiche insieme a eventuali promozioni e medaglie. All'inizio di ogni scenario possiamo assegnare a ognuno degli aerei "amici" presenti uno di questi piloti o equipaggi, e la loro bravura influirà sulle sorti dello scontro. Non è infrequente, però, vedere uno degli aerei della nostra scorta, al quale abbiamo magari assegnato un vero asso, faticosamente "costruito" in molte missioni, inesorabilmente abbattuto dai piloti nemici. Può causare un certo disappunto, ma così è la guerra...

Dopo una breve pausa per sintonizzare la radio di bordo, che costituisce, tramite la solita "ruota dei codici", la sola protezione del programma, siamo finalmente in volo (o sulla pista, pronti a un facile decollo). Un'occhiata alla mappa, che in ogni momento ci fornisce la situazione (aggiornata ogni due minuti) degli obiettivi in aria o al suolo, e ci tuffiamo nella nostra prima missione. La grafica del gioco, praticamente identica a quella del suo predecessore, è uno strano connubio di grafica vettoriale e di immagini pre-disegnate. Il primo tipo di grafica viene utilizzato per rappresentare il paesaggio e gli obiettivi terrestri al suolo, mentre il secondo è usato per gli aerei, gli obiettivi navali e gli effetti speciali quali esplosioni, fumate e ammaraggi. Le immagini disegnate sono necessariamente in numero limitato e mentre ciò non provoca problemi nel caso di velivoli distanti (anche se, in qualche caso, è difficile capire in che direzione sta andando l'aereo osservato), si nota una certa imprecisione nel disegno e nel movimento a scatti nel caso di "incontri ravvicinati" con il nemico. L'azione, però, è così frenetica che dopo un po' non ci si fa più caso.

La guida del nostro velivolo è abbastanza semplice e i comandi abbastanza completi, anche se, ovviamente, non si può parlare di una vera simulazione. *Their finest hour* è innanzitutto un gioco, e tale rimane. La precisione, più che nella simulazione di volo, è da ricercare altrove. Innanzitutto nel sonoro, decisamente eccezionale, almeno per chi possiede un megabyte di memoria: gli effetti sonori e soprattutto il rumore dei motori, diverso per ogni apparecchio, sono tremendamente realistici, tanto da far provare a volte qualche brivido quando il caccia che ci si è messo in coda comincia a sforacchiare il nostro bombardiere.

Al primo impatto con il programma è difficile non essere entusiasmati dalle sensazioni del "dogfight": i piloti nemici sono decisamente abili, ed è difficile, almeno nelle prime partite, scollarsi dalla coda un "Top Ace" come Werner Molders. L'atmosfera che personalmente preferisco, però, viene dalle missioni di bombardamento: il cupo rombo dei motori di un He 111, il lungo trasferimento volando sopra la Manica, il frenetico spostarsi da una posizione di mitragliere all'altra nel tentativo di allontanare gli Hurricane sfuggiti alla nostra scorta, l'attento calcolo del punto di sgancio delle bombe mentre tutto intorno è un vero e proprio inferno di fuoco, e infine il lento ritorno alla base con l'aereo in fiamme. Sono tutte sensazioni magistralmente rese da questo programma, e ne costituiscono il vero punto di forza.

Sono presenti, inoltre, altre caratteristiche di tutto rispetto, prima fra tutte la possibilità del replay. In ogni momento è possibile rivedere il materiale registrato, cambiando magari punto di vista e con la possibilità del "fermo immagine". A conclusione della missione vengono tirate le somme sull'esito della stessa e vengono aggiornati i file relativi ai vari piloti che vi hanno partecipato, comprese eventuali medaglie e promozioni. Il programma presenta inoltre un'opzione per rivedere comodamente questi file personali, insieme alla classifica dei dieci migliori piloti. Purtroppo, è assente un'opzione che permetta di rimuovere i piloti non più desiderati dal disco. A meno di non riutilizzare sempre gli stessi nomi, il disco dati si riempirà rapidamente di una moltitudine di cadaveri e prigionieri di guerra che aumenteranno notevolmente i tempi di accesso allo stesso. Unico grosso neo del programma sono infatti i soporiferi tempi di caricamento. Mentre il "disk swapping" è fortunatamente abbastanza ridotto, quasi a ogni selezione di menu si dà il via a un processo di "grattugiamento" delle testine del drive, che in alcuni casi può durare anche più di un minuto. A parte questo difetto, però, *Their Finest Hour* si rivela un programma con qualcosa per tutti: piacerà agli amanti della ricostruzione storica per la sua accuratezza, a quelli del combattimento aereo per l'emozione che è in grado di fornire, e a quelli delle simulazioni in genere per il suo notevole realismo e la grande cura in ogni particolare.

A.P.T. ■

PROGRAMMARE L'AMIGA

PROGRAMMARE L'AMIGA VOL. I

Per programmare l'Amiga in C e in Assembly

Un manuale di programmazione molto approfondito sulle funzioni e sulle strutture che l'Amiga mette a disposizione per la grafica, l'animazione e la gestione multitasking del sistema. Aggiornato, conciso e organizzato tenendo conto delle esigenze dei programmatori, questo volume costituisce una guida ideale per la costruzione di applicazioni che sappiano sfruttare realmente tutte le capacità dell'Amiga.

Il libro copre più di 300 funzioni di sistema, suddivise in sette argomenti principali:

- La libreria Exec
- La libreria Graphics
- La libreria Layer
- La libreria Intuition
- La libreria Icon
- Le animazioni
- La gestione dei testi

«Programmare l'Amiga Vol. I è un imponente strumento di lavoro, strutturato in modo intelligente e indispensabile per la vostra biblioteca tecnica». (Byte)

784 pagine – 18,8 × 23,5 cm
ISBN 88-7803-004-X – L. 80.000

PROGRAMMARE L'AMIGA VOL. II

La programmazione e la gestione dei dispositivi di I/O

Programmare l'Amiga Vol. II è un manuale che analizza in modo approfondito la programmazione dei dispositivi di I/O. Viene trattata anche la generazione dei suoni e la sintesi vocale. Nel volume sono inclusi moltissimi diagrammi e tavole di riferimento per illustrare i singoli concetti.

Il cuore dell'opera è comunque la descrizione di ogni dispositivo, la sfera dei suoi possibili impieghi, la sua programmazione e la sua gestione attraverso i linguaggi C e Assembly.

- I dispositivi di I/O • La gestione dei dispositivi
- I dispositivi Audio • Narrator • Parallel • Serial
- Input • Console • Keyboard • Gameport
- Printer • Clipboard • Timer • TrackDisk

528 pagine 18,8 × 23,5 cm
ISBN 88-7803-005-8 – L. 70.000

I libri IHT sono disponibili nelle migliori librerie e computer shop. Per ordini diretti servirsi del modulo pubblicato a pagina 95
IHT Gruppo Editoriale – Via Monte Napoleone, 9 – 20121 Milano – Tel. 02/794181-794122 – Fax 02/784021 – Telex 334261 IHT I
Distribuzione: RCS Rizzoli Libri – Via Scarsellini, 17 – 20161 Milano – Tel. 02/64068508

SALONI ITALIA

LA COMMODORE SI PRESENTA ALLO SMAU '90

Tutto lo stand dedicato al nuovo Amiga 3000 e "invaso" dagli indipendenti. Niente anteprime e niente CDTV... ma abbiamo scovato il portatile MS-DOS compatibile 286 LT

di Giovanni Varia

Dal 4 all'8 ottobre, alla Fiera di Milano, si è tenuto il più importante appuntamento italiano con l'informatica: lo SMAU, rassegna che ormai ha assunto dimensioni davvero ciclopiche (occupa quasi l'intero quartiere fieristico) e risonanza a livello europeo (961 espositori e oltre 180 mila visitatori).

L'immancabile presenza della Commodore Italiana era caratterizzata dai soliti due elementi: la galleria-giochi per i visitatori e le dimostrazioni degli indipendenti (ossia aziende diverse dalla Commodore Italiana). La parte del leone era affidata all'Amiga 3000, mentre non si è vista traccia dell'atteso CDTV, la cui uscita sembra essere stata definitivamente posticipata al febbraio/marzo '91.

Anche se sono anni che lo scriviamo, evidentemente non siamo ascoltati: secondo noi, un appuntamento come lo SMAU non può essere affidato praticamente per intero alle dimostrazioni dei prodotti importati o realizzati da varie aziende indipendenti dalla Commodore!

Chi si avvicina allo stand ha senza dubbio il desiderio di trovare riunita l'intera gamma dei prodotti presenti nel listino, di poterli provare e di trovare personale preparato in grado d'illustrarli e di rispondere alle domande di carattere tecnico. Affidare la propria presenza alle presentazioni di indipendenti "più o meno capaci", a una galleria-giochi per ragazzini maniaci del videogame (e della folla e del sudore) e non mostrare l'intera gamma in listino è secondo noi un errore. Sono anni che lo scriviamo su questa rivista e sono anni che i nostri lettori ci manifestano questa stessa opinione... ma alla Commodore Italiana evidentemente ritengono che sia noi, sia i nostri lettori siamo in errore... Vedremo cosa ci riserverà il prossimo anno.

Con questo non vogliamo comunque assolutamente sminuire la validità dei prodotti di molti degli indipendenti presenti a questa edizione dello SMAU. Anzi, non indugiamo oltre e passiamo con decisione subito a loro.

Le dimostrazioni degli indipendenti

Michele Console Battilana della **Cloanto**, sempre serio e professionale, era presente con il suo cavallo di battaglia: *C1-Text* (che all'estero si chiama *Personal Write*), di cui veniva mostrata la nuova versione 3.0. Le due novità più importanti della 3.0 sono il driver Postscript e il nuovo manuale. Quest'ultimo ha il 50% di pagine in più, numerose illustrazioni e una sezione iniziale contenente le 25

domande classiche del neofita. Inoltre, sono state apportate al programma oltre duecentocinquanta modifiche, in gran parte legate a nuove funzioni. Una cosa di cui Battilana andava molto fiero era il fatto che *C1-Text 3.0* è stato uno dei primissimi programmi a ricevere dalla Commodore International la certificazione di compatibilità con l'Amiga 3000 e il sistema operativo 3.0, pur rimanendo perfettamente funzionante anche con il vecchio sistema operativo e i modelli di base anche con soli 512K. Il prezzo al pubblico di *C1-Text 3.0* è di 89.000 lire. È anche disponibile un kit di upgrade per gli utenti registrati della versione precedente, i quali non si devono preoccupare di niente: riceveranno una lettera dalla Cloanto che offrirà la nuova versione a 25.000 lire. Tra gli altri prodotti presentati con il marchio Cloanto, ricordiamo anche *Personal Fonts Maker* e un analizzatore di testi ASCII (quante volte ricorre la tal parola...) destinato a uso scolastico.

La **C.B. Music** aveva una novità computermusicale marcata Kawai: la tastiera FunLab completa d'interfaccia MIDI e di software dedicato all'Amiga. Sessantuno tasti, 14 voci, 100 timbri e 100 ritmi sono le caratteristiche principali di questa tastiera a suoni campionati, il cui prezzo è di 735.000 lire (Iva compresa). A un prezzo non elevatissimo si può quindi suonare e far registrare e poi riprodurre il tutto dall'Amiga, o inserire la musica direttamente dalla tastiera del computer e utilizzare la tastiera per l'esecuzione. Per chi volesse spendere meno è anche disponibile la più economica tastiera Kawai MS710 (49 tasti, 10 voci, 24 timbri, 24 ritmi) a 375.000 lire (Iva inclusa).

Sempre sul fronte musicale era presente anche la **Soundware**, azienda che distribuisce nei computer shop la linea di prodotti di computermusica della Roland (gli expander CM32, CM64...) che abbiamo ampiamente recensito sul numero 4/90 di *Commodore Gazette*. In particolare, veniva proposta l'accoppiata con l'Amiga 3000, messo in luce come la workstation ottimale per la produzione di musica sincronizzata a immagini in spot, jingle, videoclip, filmati... La Soundware ha colto l'occasione dello SMAU anche per presentare la sua nuova Divisione Multimediale, che si occuperà sia della commercializzazione di prodotti, sia dell'offerta di servizi. Questa società (non nello stand Commodore, ma in quello adiacente occupato dalla Leader Distribuzione) presentava anche prodotti software (*Music-X, KCS 3.0, Copyist, Tiger Cub*), hardware musicale (interfaccia MIDI Model A, generatore/lettore di codice SMPTE Phantom) e hardware video (genlock Minigen frame grabber Vidi Amiga e VideoVedi per la desktop presentation).

Da sopra a sotto: Michele Console Battilana della Cloanto presenta C1-Text 3.0. La tastiera Kawai FanLab importata dalla C.B. Music. La Soundware nello stand della Leader

Nel campo della computer animation, dallo schermo di un Amiga 3000 si potevano vedere le meraviglie dello **studio MGM** (vincitore del noto concorso Bit Movie). Mentre ammiravamo una scacchiera tridimensionale, Mrsek Giuseppe Milko ci spiegava che lui e i suoi collaboratori realizzano animazioni tridimensionali, computergrafica e loghi per applicazioni multimediali e la pubblicità. Per il loro lavoro gli strumenti principe sono un Amiga 3000 e un Amiga 2000 con scheda della Great Valley Products equipaggiata con Motorola 68030.

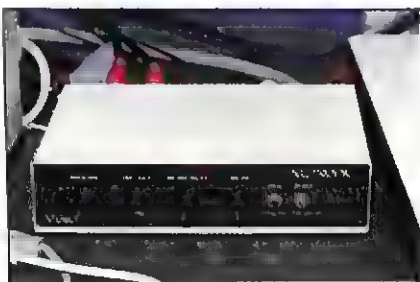
Mario Bucolo della Bucolo Computer Graphics di Catania rinnovava la sua ormai tradizionale presenza allo SMAU annunciando che la sua ditta era confluita nella società di recente costituzione denominata **ABIS**. La ABIS era presente allo stand Commodore con i seguenti prodotti: Windovid, un dispositivo per Amiga, e anche per macchine MS-DOS che applicato dietro alla vetrina di un negozio permette a chi opera dal lato esterno di attivare i tasti e quindi di comunicare con un computer; Infoself, una stazione informativa basata sul sistema Windovid che può essere anche blindata e collocata di conseguenza in luoghi pubblici; Catania Viva, un sistema informativo grafico realizzato con l'Amiga che con le sue oltre 400 immagini e scritte in tre lingue è stato installato dall'Azienda Provinciale per il Turismo di Catania nei punti chiave della città; *Totolive* e *Elezlive*, due programmi per Amiga, venduti ormai a una ventina di emittenti televisive, che permettono la gestione in tempo reale dei dati di campionati di calcio e di consultazioni elettorali (mantenendo sempre il computer in diretta).

Corsi di formazione su Amiga personalizzati e collettivi (AmigaDOS di base, 18 ore, 340.000 lire; DTP di base, 18 ore, 450.000 lire; DTP avanzato, 36 ore, 1.260.000; Videografica di base, 18 ore, 480.000 lire; Videografica avanzato, 36 ore, 1.590.000 lire - prezzi Iva esclusa), computergrafica e produzione video per la televisione e la pubblicità, realizzazione di programmi personalizzati, vendita con instal-

lazione, assistenza e personalizzazione di sistemi completi per il desktop publishing (basati su Amiga 3000) o per la videotitolazione... Questo è il ventaglio di offerte di un'altra società che era presente all'interno dello stand della Commodore Italiana: la **Digimail**. Paola Pellegrini non ha mancato d'illustrarci con orgoglio la postazione della società milanese, che era davvero ben nutrita di genlock, frame grabber, stampante a getto d'inchiostro e altro ancora... Tra questo ben di Dio, veniva illustrata al pubblico parte della loro attività. Paola ci ha anche condotto al punto informativo Commodore costituito da un videocatalogo dei prodotti CBM, realizzato ovviamente sempre dalla eclettica Digimail (forse visto il ventaglio di attività la parola *mail* comincia a non essere più tanto appropriata...).

Prenotazione dei biglietti dei concerti in tempo reale? Presentazione delle varie date con schermate grafiche e brani di musica campionata (rigorosamente del gruppo o del solista che danno il concerto)? Questo e altro è stato realizzato su un Amiga 3000, che diventa così uno strumento destinato a tutti i punti vendita nei quali si effettuano le prenotazioni di spettacoli musicali. Questa originale applicazione (Italian Music System) è stata realizzata dalla **O.T.S.** di Roma, per la quale era presente allo SMAU Antonino Ocello.

Tra tutti gli indipendenti, chi ci ha accolto nel modo più originale è stato Adolfo Fogli della **Sinetica**. Alla domanda: «Lei cosa presenta, cosa offre?» ci è stato risposto: «Io non offro e non presento proprio niente!». Allorché inevitabile è stato il nostro: «Ma allora lei cosa ci fa qui?» seguito dalla fatidica risposta conclusiva: «Non lo so neanch'io. La nostra è un'azienda che si occupa di ricerche di mercato, e nelle presentazioni ai nostri clienti ci avvaliamo d'immagini realizzate con un Amiga 2000 trasferite su diapositiva. Se la Commodore ci offrirà un A3000 con un buono sconto cambieremo il nostro A2000, altrimenti continueremo a usare il vecchio modello. Comunque, l'unica affermazione che credo possa interessarla è questa: per l'uso che ne facciamo siamo soddisfatti dell'Amiga. Tutto qui. Non sappiamo bene neanche noi perché la Commodore ci abbia invitati a partecipare...».

Perché non usare l'Amiga 3000 e *AmigaVision* per realizzare un videocatalogo con il quale le agenzie di viaggio possano presentare in modo diverso dalle consuete brochure le loro offerte ai clienti? A questo ci ha pensato la **Digiteam**. Mentre ci veniva mostrata una possibilità di vacanza nell'isola di Bali, ci veniva ovviamente fatto presente che il videocatalogo viene personalizzato a seconda delle esigenze dell'agenzia. Oltre a questo, Eugenio Ciceri della Digiteam ci ha anche mostrato un'altra creatura dell'azienda milanese: un nuovo programma di disegno battezzato *Micropaint*.

Da sopra a sotto: Mario Bucolo e il suo sistema informativo applicato allo stand Commodore. Videon III della Newtronic distribuito dalla Curiotré. La postazione Digimail

I genlock per Amiga dell'australiana Neriki erano presentati dal loro distributore italiano, la **E.C.M.** I modelli disponibili sono l'Image master GL1187C (broadcast, con anche uscite Y-C, 4.100.000 lire Iva esclusa) e il Desk Top GL1189 (broadcast, 2.300.000 lire Iva esclusa).

Videon III è un videodigitalizzatore per l'Amiga (è disponibile anche per PC AT). È possibile digitalizzare da telecamera e videoregistratore (segnale videocomposito oppure S-VHS). Il digitalizzatore acquisisce immagini in 24 bit, 8 bit per componente, permettendo di arrivare fino a 16.777.216 colori... Vi interessa? Videon III (480.000 lire Iva esclusa), che veniva presentato anch'esso all'interno dello stand Commodore è prodotto dalla Newtronic e distribuito dalla **Curiotré**.

L'ultimo nostro incontro con un indipendente è stato con la **Logitek**, che propone soluzioni complete per il desktop publishing, e, attraverso la Videographic, servizi di grafica (impaginazione, animazione, creazione loghi, titolazione, riversamento passo 1, editing RVM, realizzazione videovetrine) e di produzione video (filmati industriali e pubblicitari con Betacam...).

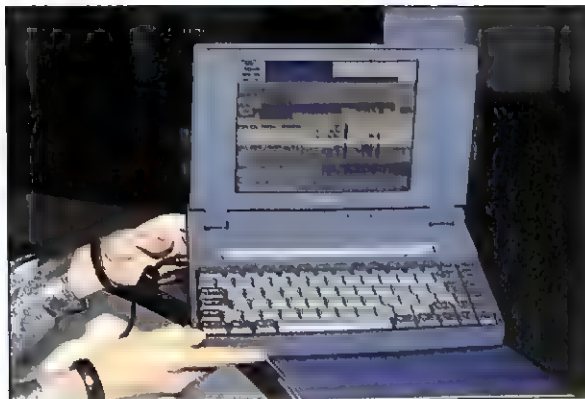
Lo stand C.T.O.

A differenza degli anni passati, per l'edizione '90 dello SMAU la C.T.O. di Bologna ha scelto di allestire uno spazio vicino, ma al di fuori dello stand della Commodore (nello stand Commodore era presente solo una postazione). Franca Tommasi ci ha gentilmente ragguagliato sulle più recenti novità. Particolare importanza riveste un nuovo accordo di distribuzione esclusiva per l'Italia: l'azienda bolognese ha infatti firmato un contratto con la Lucas-Film, e quindi tutti i titoli da *Secret of the monkey island* in avanti avranno il marchio C.T.O.

Da una joint-venture con la francese Coktel Vision è nata la C & C Education, società che si occuperà della diffusione di prodotti educativi su tutto il territorio nazionale, per ogni ordine di studi, in versioni Amiga e MS-DOS. I prodotti già disponibili sono sedici, divisi nei temi: inglese, matematica, ripassi e logica. Sono in programmazione anche titoli per materie come francese, spagnolo, tedesco, scienze naturali, geogafia... Il software è studiato per gli studenti della scuola elementare e media, ma anche per i più piccini, e si

avvale di divertenti metodologie di autoapprendimento che ci sono state illustrate, e appaiono interessanti.

Sempre allo stand C.T.O. abbiamo incontrato anche Roberto Ascione, il giovane general manager della napoletana **Sky Softworks** i cui prodotti rigorosamente per Amiga sono distribuiti dalla C.T.O. La filosofia di questa azienda è quella di avvalersi nelle sue produzioni di routine d'intelligenza artificiale. Il primo programma che ha beneficiato di criteri di IA è stato l'ormai disponibile *I.M.A.G.E.* per la gestione delle oreficerie. Tra le funzioni di questo pacchetto citiamo: gestione magazzino, contabilità, anagrafico clienti, emissione documentazione fiscale... In fase di realizzazione sono invece *W.O.M.* per reparti medici ospedalieri, due programmi per video multimediali e un sistema di segreteria automatizzata (*Eyes*). Il progetto forse più ambizioso è invece *Life*, un simulatore accelerato delle funzioni del fegato umano che a detta della Sky Softworks consentirà prodezze come la sperimentazione di farmaci (siamo proprio curiosi...).

Roberto Ascione ci ha infine illustrato un altro programma per Amiga che è invece già disponibile: *Met.A.* Si tratta di un pacchetto che è in grado di annullare completamente la necessità di usare carta e penna nell'organizzazione di un congresso, il che vuol dire che il software è orientato a dare un ausilio nella risoluzione di problemi sia creativi sia pratici (prezzo a partire da 1.700.000 lire, a seconda delle personalizzazioni richieste).

Sopra: Roberto Ascione della Sky Softworks illustra Met.A. Sotto: l'unica anteprima che abbiamo scovato, il 286 LT

La Leader Distribuzione con la Soundware

Sempre nei pressi dello stand Commodore era presente anche la Leader, l'importante distributore di Casciago. Oltre a far mostra di parte dell'imponente catalogo di software disponibile (del quale non facciamo menzione, perché le novità, con cadenza settimanale, sono tante e tali che occuperebbero troppo spazio per essere citate tutte), la Leader aveva anche riservato una parte del suo stand alla Soundware che aveva così un ulteriore spazio per la presentazione dei prodotti di cui vi abbiamo già parlato.

Arrivederci al prossimo anno: qualcosa cambierà?

La nostra panoramica sulle novità legate al mondo Commodore presente allo SMAU '90 è terminata. Purtroppo per i nostri lettori possessori di C-64/128, alla fiera non abbiamo trovato proprio nulla di nuovo che potesse costituire per loro motivo d'interesse.

Chissà poi se per il prossimo anno saranno ascoltati dalla Commodore Italiana i nostri consigli di dare dimostrazioni dell'intera gamma in listino...

Vogliamo comunque concludere con una curiosità: anche se anteprime allo stand Commodore non se ne sono viste e anche se il CDTV è ancora da venire, il nostro occhio vigile un'anteprima l'ha scorta: un'avvenente signorina seduta dietro a uno dei banconi dai quali si distribuiva documentazione al pubblico aveva dinanzi a sé un bel portatile che a prima vista sembrava un Toshiba. Non era però un Toshiba, bensì una nuova creatura sulla quale la serigrafia diceva **Commodore 286 LT**. ■

LE NOVITÀ DELL'IBTS

Grande affluenza di visitatori allo stand Commodore in occasione dell IBTS (18-22 ottobre, fiera di Milano/Lacchiarella), la fiera internazionale dedicata al broadcasting. Doverosa precisazione per chi non conosce questa manifestazione è la segnalazione della partecipazione di grandi nomi del mondo video come SONY e JVC, che esponevano le loro ultimissime novità. Roba da far girare la testa anche ai rappresentanti delle molte TV private venuti da tutt'Italia per scoprire le ultime tendenze nel campo video, che per inciso vedono una decisa spinta verso i sistemi in cui il segnale viene portato sui due canali separati di luminanza e crominanza crescita quindi di Super VHS M2, Hi-8 e Betacam, e diminuzione sui 3/4 di pollice.

Allo stand Commodore che finalmente quest'anno si è decisa a partecipare a una manifestazione specifica di un settore in cui l'Amiga potrebbe fare la parte del leone, erano presenti sei espositori indipendenti dalla Commodore, con riproposte (le stesse dello SMAU) e novità tutte basate sull'Amiga 3000.

La Digimail di Milano ha partecipato con due postazioni. La prima era una soluzione studiata per rappresentare una risposta integrata per le TV private completa di software per la titolazione e di due programmi "per la diretta". Il programma di titolazione proposto era *Professional Titler* per il quale la Digimail ha raggiunto un accordo di

distribuzione. I due pacchetti di software televisivo erano invece *SportShow* e *SportPlay* (entrambi sono attualmente già in uso presso alcune emittenti commerciali). La seconda postazione Digimail era rappresentata dall'Info Point, già presentato in SMAU, arricchito con una piantina interattiva dello stand

La ECM presentava una postazione dedicata alla videografica (con *DPaint*, *Digipaint* e altri noti programmi), mentre la Soundware proponeva la soluzione per la sincronizzazione audio/video, possibile sia con registratori professionali che amatoriali.

La videografica tridimensionale era presentata dalla Logitek, che proponeva il conosciuto software *Caligari* con editor prospettico per la creazione di oggetti semplici e complessi, con rendering a 32.000 colori ottenuto tramite l'utilizzo di una scheda Truevision Targa 16 e di una Bridgeboard.

Altri "pezzi" mai visti prima dal pubblico italiano erano un prototipo austriaco di Frame Grabber/Buffer a 24 bit presentato dalla Quest, che speriamo venga sviluppato con l'attenzione che merita, il Frame Talk della Desktop Video di Milano e il DVE-10 della tedesca Videocomp.

Il Frame Talk (altra bella realizzazione italiana) è un sistema completo hardware/software che permette di effettuare montaggi video utilizzando il computer. L'Amiga è interfacciato direttamente con un lettore video, un videoregistratore e una centralina di montaggio In questo modo l'operatore può creare a video uno storyboard d'immagini digitalizzate definendone ordine, durata e transizione. Quindi lo storyboard può essere riversato su videoregistratore L'interessante sistema viene commercializzato direttamente dalla SONY (tel. 02/6128221) e si interfaccia per ora ai VCR previsti per l'interfaccia seriale (RS 232 e RS 422), mentre è in fase di studio una versione per la parallela.

In campo amatoriale ha suscitato interesse il DVE della VIDEOCOMP, una ditta tedesca che viene attualmente rappresentata in Italia dalla FUMEO (tel. 02/2551651). Il prodotto unisce insieme un genlock dalle buone caratteristiche, un generatore digitale di effetti speciali (per ora limitati al Picture in Picture) e un digitalizzatore molto veloce.

Infine, va segnalata la presenza, al di fuori dello stand Commodore, della Telav International (tel. 02/4455741), importatore della Magni, con il nuovissimo genlock "serie 4000" per Amiga 3000 alla sua prima uscita in Italia, che si conferma come il miglior genlock disponibile sul mercato. La scheda mantiene inalterate le sue caratteristiche e, oltre a funzionare perfettamente sul 3000, possiede uno switch per commutare il segnale da A3000 ad A2000 e viceversa

Speriamo che questa presenza della Commodore a una fiera specializzata sia uno degli anelli di una lunga catena, che è diventata ormai indispensabile per affermare un'immagine di reale affidabilità del marchio Commodore presso un'utenza professionale. ■

Per ulteriori informazioni contattare direttamente:

ABIS srl
Via A. Rampi, 2a
95020 Aci S. Filippo (CT)
(Tel. 095/870675-6 - Fax 870676)

C.B. Music
Via Padova, 39
20127 Milano
(Tel. 02/2895022-26140160)

Cloanto Italia
Via G.B. Bison, 24 545902
33100 Udine
(Tel. 0432/46612 - Fax 609051)

Commodore Italiana spa
Viale Fulvio Testi, 280
20126 Milano
(Tel. 02/661231 - Fax 66101261)

C.T.O. srl
Via Piemonte, 7/F
40069 Zola Predosa (BO)
(Tel. 051/753133 - Fax 753418)

Curiotré srl
Cavaria (VA)
(Tel. 0331/212585)

Digimail srl
Via Coronelli, 10 - 20146 Milano
(Tel. 02/426559-427621 - Fax 427768)

Digiteam
Via Gustavo Modena, 9
20129 Milano
(Tel. 02/29409073)

E.C.M. srl
Centro Commerciale Giada
Via dei Platani, 6
20020 Arese (MI)
(Tel. 02/9386084 - Fax 9380710)

Leader Distribuzione srl
Via Mazzini, 15
21020 Casciago (VA)
(Tel. 0332/212255)

Logitek/Videographic
Via Golgi, 60 - 20133 Milano
(Tel. 02/2666274 - Fax 231290)

MGM Studio
Via G. Verdi, 6
25079 Vobarno (BS)
(Tel. 0365/598757)

O.T.S. Electronics
Via dei Gelsi, 130/b
00172 Roma
(Tel. 06/2597209)

Sinetica sas
Via Mauro Macchi, 38
20124 Milano
(Tel. 02/66983848
Fax 66983885)

Sky Softworks
Via M. da Caravaggio, 89
80126 Napoli
(Tel. 081/7142124)

Soundware srl
Via Mazzini, 12
21020 Casciago (VA)
(Tel. 0332/222052
Fax 228288)

AMIGA 3000: IMPRESSIONI D'USO

Un reportage dalla tastiera del nuovo gioiello della famiglia Amiga

di Alfredo Prochet

Pur essendo stati abituati al suo aspetto dal gran numero di fotografie, schemi e disegni in circolazione dal momento della sua presentazione ufficiale, le dimensioni e l'eleganza dell'A3000 "visto dal vivo" non possono ugualmente fare a meno di stupire: l'unità centrale appare compatta in maniera quasi incredibile, specialmente pensando a tutto quello che contiene. In un certo senso, addirittura, l'ottimo monitor 1950 e la tastiera, pressoché identica a quella dell'Amiga 2000, appaiono quasi sproporzionati in confronto al resto, specialmente la seconda, più larga dell'unità centrale di dieci centimetri buoni, ma ovviamente è solo una questione di abitudine: pochi minuti "per fare amicizia" e ci si sente perfettamente a proprio agio.

Uno sguardo al manuale fornito: si tratta di un bel manuale ad anelli come quello che accompagnava l'Amiga 1000, contenente la documentazione relativa all'Amiga 3000 e al *Workbench 2.0*, inclusa l'intera descrizione del linguaggio ARexx. Sono compresi nella confezione anche sei dischetti: *Workbench* ed *Extras 1.3*, il disco denominato *SuperKickstart* contenente *Kickstart 1.3* e *2.0*, *Workbench* ed Extras *2.0* e un disco d'installazione, funzionante solo sotto *2.0* ma valido per entrambi i sistemi, che comprende tutti i necessari programmi per la gestione dell'hard disk.

Proseguendo in un primo esame esterno, si fa subito notare il pulsante di accensione. Personalmente, non ne approvo molto la collocazione frontale, che dovrebbe comunque rivelarsi abbastanza protetta da evitare tragiche pressioni accidentali, anche se, con ogni probabilità, non sarà a prova del solito amico "furbo" che vuol fare uno scherzo o di quello esasperato dall'aver perso la dodicesima partita a *Kick Off* di fila.

Anche il drive da 3,5" si trova in una posizione esteticamente gradevole, incorporato com'è nel frontale stesso, senza essere incorniciato da antiestetiche fessure. Peccato però che il led di attività del drive sia, almeno per quanto riguarda l'unità fornita con la macchina, quasi completamente invisibile in un ambiente normalmente illuminato. Per sapere se il drive è attivo, a volte, è necessario appoggiare un dito sul dorso del floppy inserito, per poter avvertire le vibrazioni causate dagli eventuali spostamenti della testina, il cui movimento, infatti, è quasi inudibile. I led di accensione e dell'hard disk sono invece adeguati, specie se confrontati con quei fanali abbaglianti montati sull'A2000.

I connettori per il mouse e il joystick sono finalmente di nuovo in una posizione accessibile, così com'erano sul 1000, sul lato destro della macchina, insieme a quello della tastiera. Sul retro, invece, tre novità: il connettore SCSI a 25 pin (ve ne è anche uno interno a 50 pin sulla piastra madre), il connettore a 15 pin tipo VGA per l'uscita video a 31 KHz e infine l'interrutore che consente di attivare o disattivare in qualunque momento il VDE: il Video Display Enhancer, controllato dal chip Amber, in grado di eliminare l'interlace.

Partenza

Sebbene l'Amiga 3000 a nostra disposizione sia il più "lento" della serie, quello a 16 MHz, l'impressione di velocità che si ha fin dai primi istanti è eccellente, merito in gran parte da attribuirsi anche all'hard disk montato, uno dei più veloci e silenziosi che abbia mai incontrato (per la cronaca, il programma che si occupa della formattazione a basso livello ci informa che si tratta di un HD Quantum P40S). Esso viene fornito già diviso in tre partizioni: una da 6 MB contenente il sistema 2.0, una da 2 MB contenente il sistema 1.3 e una di lavoro che occupa lo spazio rimanente, nel nostro caso circa 32 MB. Le partizioni sono tutte gestite dal nuovo FastFileSystem in quanto, finalmente, non è più obbligatorio l'uso del vecchio FileSystem nemmeno per eseguire il boot.

Terminato il brevissimo processo d'inizializzazione del sistema, ci troviamo finalmente nello schermo del *Workbench 2.0*. Le novità sono tali e tante che è difficile scegliere da che parte cominciare, l'unica cosa che appare immediatamente chiara è che sarà difficile scindere la prova del nuovo software da quella della nuova macchina. Dire che sembrano fatti l'uno per l'altra può sembrare un po' eccessivo, ma in effetti non è lontano dal vero: il *Kickstart 2.0* è, o sarà, disponibile anche sugli altri modelli di Amiga, certo, ma è solo sul 3000 che dimostra veramente tutte le sue potenzialità. Un primo sguardo allo schermo e già diventano evidenti le meraviglie di cui è capace Amber insieme a un monitor multiscan. Lo schermo 640 x 512, eliminato il flickering, e grazie anche all'azzeccatissima scelta di colori, diventa infatti veramente nitido e riposante, ed è realmente difficile tornare poi a un *Workbench* non interlace senza un sospiro di rimpianto.

Essendo il sistema 2.0 già diffusamente trattato in un altro articolo in questo stesso numero della rivista, ci limiteremo a mettere in evidenza le eventuali particolarità che possono essere riscontrate sull'A3000. Una di queste la troviamo subito nella barra orizzontale con il titolo dello schermo. Di seguito al nome "Workbench 2.0", il valore indicante la quantità di

memoria libera è stato spezzato in due: in memoria chip, "graphics mem", e in tutto quello che memoria chip non è, "other mem". È proprio da quest'ultimo valore che viene una piccola sorpresa: mezzo megabyte di memoria fast risulta mancante all'appello (ricordiamo che il sistema viene fornito con un megabyte di memoria chip e uno di fast) e la memoria totale del sistema diventa quindi di soli 1,5 megabyte. E l'altra memoria, chi se l'è rubata? La risposta è particolarmente semplice: il *Kickstart*, che non essendo ancora nella sua versione definitiva, non è realmente presente sotto forma di chip ROM ma viene caricato dall'hard disk all'accensione del sistema in maniera del tutto trasparente per l'utente. Parte della memoria fast viene riconfigurata all'indirizzo opportuno, $F80000, e tutto funziona a meraviglia. In questo modo diventa possibile scegliere con estrema facilità quale versione di sistema operativo utilizzare, se 1.3 o 2.0, e sarà possibile nello stesso modo passare ad aggiornamenti e versioni successive senza alcun intervento "cruento" sull'hardware. La scelta di mantenere un *Kickstart* "variabile" si era già rivelata ottima sull'Amiga 1000, dove, per una simile versatilità, occorreva pagare il modico prezzo dell'inserimento di un disco in più all'accensione e di una ventina di secondi di caricamento. Sull'Amiga 3000 non occorre più fare nemmeno questo: il procedimento è del tutto automatico e pressoché inavvertibile in termini di tempo. Il fatto che 512K di memoria, nel caso della versione 2.0, vengano dedicati a questo scopo, non influisce più di tanto sulle prestazioni del sistema. Intendendo con questo che, su una macchina come l'A3000, un'espansione di almeno 4 MB appare comunque un *sine qua non*.

Desiderando caricare il *Kickstart 1.3* dall'hard disk oppure una nuova (o vecchia) versione direttamente da dischetto, è sufficiente tenere premuti entrambi i tasti del mouse all'accensione (attenzione: non al reset) del sistema. Comparirà in tal caso una schermata contenente un menu tramite il quale è possibile accedere con facilità alla versione desiderata. L'unica perplessità viene destata dal fatto che, di tutta questa procedura, il manuale fornito con la macchina non fa assolutamente menzione.

Un'altra cosa degna di nota è il fatto che le due partizioni di sistema dell'hard disk, 2.0 e 1.3, non sono accessibili contemporaneamente, ma si escludono invece a vicenda: è presente l'una oppure l'altra. Non si può dire che questo rappresenti un vero problema per l'utente, dal momento che più di una versione del sistema per volta non può comunque utilizzare; il lato singolare, però, riguarda il fatto che i programmi forniti sul disco d'installazione tentano di accedere alle due partizioni nello stesso tempo, e pertanto non funzionano o funzionano solo parzialmente.

Il VDE e le nuove risoluzioni

Una volta informato il sistema operativo, tramite il programma AddMonitor, del fatto che abbiamo collegato un monitor multiscan, in particolare il 1950, del quale non posso che tessere le lodi, siamo pronti per iniziare a sperimentare, con il programma *WBScreen*, gli effetti delle varie risoluzioni. Il VDE lavora in modi diversi a seconda che ci si trovi o meno in modo interlace. Nel primo caso (interlace attivo), Amber visualizza ogni quadro video insieme a quello immediatamente precedente, in modo da evitare il fastidioso flickering che tutti ben conosciamo. Nel caso di animazioni molto veloci si può verificare un certo "effetto fantasma" causato dal fatto che metà dell'immagine visualizzata è in ritardo di 1/25 di secondo. Niente del genere accade invece nei modi video non-interlace. In questo caso, infatti, il VDE si limita a visualizzare ogni riga due volte, in modo da eliminare il distacco fra le varie linee e fornire un display nitido e compatto.

In alta risoluzione i cambiamenti sono davvero eccezionali. È veramente facile non accorgersi del fatto che il *Workbench* utilizza uno schermo 640 x 512, tanto l'immagine appare naturale. Strano ma vero, questa stessa caratteristica a volte diventa persino eccessiva in bassa risoluzione: i pixel sono definiti infatti con tale precisione da sembrare dei mattoncini del LEGO, e le zone dove il contrasto fra i colori è maggiore appaiono a volte un po' troppo spigolose e prive di quell'antialiasing naturale fornito dal leggero arrotondamento tipico dei comuni monitor. La qualità dell'immagine è resa evidente anche dalle sue caratteristiche cromatiche: sebbene in alcuni casi i colori appaiano meno "vivi" del solito, diventa finalmente possibile apprezzare anche le sfumature con valori RGB molto bassi (del tipo $121) sino a oggi, almeno a luminosità normale, quasi indistinguibili dal nero.

Ma ora è giunto finalmente il momento di mettere alla prova il chip Super Denise e di sperimentare le nuove risoluzioni disponibili. Il Productivity Mode (640 x 480) è visivamente molto simile all'alta risoluzione interlace (640 x 512) ma possiede alcune particolarità degne di nota. Innanzitutto, la sua frequenza di scansione raddoppiata (31,25 KHz PAL) viene generata direttamente dal chip Super Denise, e non dal VDE, che si disattiva automaticamente quando viene scelta questa risoluzione. Ciò porta ad alcune conseguenze non del tutto piacevoli: l'immagine sul monitor, regolato per ottenere un display centrato con il VDE inserito, si sposta sensibilmente, ed è necessario intervenire manualmente sui controlli del 1950 per centrarla nuovamente. In modo interlace, inoltre, un comunque visivamente assurdo 640 x 960, il buon vecchio flickering torna a distruggere le nostre retine. Infine, la scelta di colori si riduce drasticamente: un massimo di 4 colori presi da una palette di 64. È soprattutto quest'ultimo valore che fa sentire maggiormente il proprio peso negativo. Gli unici vantaggi di questo modo video

rispetto all'hi-res interlace "filtrato" tramite VDE sono la possibilità di funzionare anche senza il VDE stesso (ma bisogna comunque avere l'uscita video a 31 KHz e un monitor multiscan), e l'assenza del citato "effetto fantasma" (d'altra parte, però, chi vuol fare delle animazioni a 4 colori?).

Qualche delusione viene data anche dal Superhires (1280 x 256 non interlace e 1280 x 512 interlace). Il discorso è analogo a quello fatto per il Productivity Mode, poiché anche questa risoluzione richiede il disinserimento del VDE, pena la scomparsa di tutti i pixel nelle colonne dispari, sosituiti dalla ripetizione del pixel alla loro sinistra. In modo non interlace non si hanno in realtà effetti particolarmente spiacevoli, eccetuata la ricomparsa della sottile linea nera fra una riga e l'altra, ma il rapporto fra la risoluzione orizzontale e quella verticale (5 a 1!) ne limita, a nostro giudizio parecchio, l'utilità pratica. In modo interlace, invece, sebbene il monitor regga benissimo, altrettanto non si può dire dell'occhio di chi guarda, ormai disabituatosi al flickering. Anche qui, come nel caso precedente, spostamento dell'immagine sullo schermo e limitazione nella scelta dei colori, 4 da una palette di 64. Personalmente, al posto di queste fantomatiche risoluzioni, avrei preferito il supporto degli otto bitplane di profondità, per cui il software, e in parte l'hardware, sono già predisposti (il che avrebbe portato 256 colori "veri" sullo schermo) e una palette a 24 bit anziché 12 (più di 16 milioni di colori disponibili). Attendiamo il Video Toaster.

Il Motorola 68030

Entriamo ora in uno shell, e vediamo che cosa ne pensa il sistema operativo di essere "guidato" da un 68030. L'Exec include ora tutte le necessarie routine per la corretta utilizzazione di processori diversi dal 68000, della loro eventuale memoria cache, dei modi "burst" per l'accesso alle istruzioni o ai dati e altro ancora. Tramite il comando CPU, presente nella directory c, è possibile accedere facilmente a tutte queste caratteristiche o controllarne l'impostazione corrente. Mandando invece in esecuzione un'utility della stessa Commodore, Config, scopriamo che cosa il sistema sa di se stesso:

[illegible]

Questo è più o meno quello che ci aspettavamo, compreso il fatto che metà della memoria fast, il blocco di 512K da $7F80000 a $7FFFFFF, risulta assente, essendo stato riservato al *Kickstart*. Si può inoltre notare come questi siano "veri" indirizzi a 32 bit. I programmi che fino a oggi avevano (illegalmente) usato gli otto bit più significativi dell'indirizzo, ignorati dal 68000, per memorizzarvi dei dati, cesseranno inesorabilmente di funzionare sull'Amiga 3000.

Al solito, un accurato test della velocità è quasi impossibile, in quanto i benchmark, come sempre, lasciano il tempo che trovano, dipendendo infatti da troppi fattori diversi i loro risultati. Ciò premesso, il benchmark della Lake Forest Logic dà per l'A3000 una *performance* di 4,6 rispetto all'Amiga di base, posto uguale a 1. Un'immagine generata con *Sculpt Animate 4D* è passata da 19 minuti e 35 secondi a 6 minuti e 25, e *senza* usare il coprocessore matematico.

L'aumento di velocità riscontrabile varia moltissimo da un programma all'altro e dipende in larga misura da ciò che un programma fa del tempo a sua disposizione: non si vede tanto, per esempio, in un programma d'animazione sincronizzato con la scansione del video che utilizza magari più il Blitter della CPU, quanto in quello che deve "pensare" molto o fare grandi quantità di calcoli, come un word processor o un programma di ray tracing, o in quello che funziona "al chiodo", sfruttando al massimo ogni microsecondo a sua disposizione, come un simulatore di volo.

Peccato che una quantità veramente rilevante di programmi, comunque soprattutto giochi, cessi di funzionare, quasi sempre per lo stesso motivo: violazione di privilegio. L'uso, cioè, d'istruzioni che, normali sul 68000, sul 68030 devono invece essere eseguite in supervisor mode, pena il verificarsi di una distruttiva exception. Altra grande causa del prematuro blocco del sistema e l'utilizzazione di cicli di temporizzazione dipendenti dalla velocità della CPU: tipico il loop con una moltiplicazione all'interno, eseguito magari qualche migliaio di volte, allo scopo di far passare il tempo necessario alla stabilizzazione della testina dopo uno spostamento in fase di lettura o scrittura su disco. Se "occhio e croce" il tempo è sufficiente sul 68000, con un 68030 che funziona inoltre a velocità superiore, il loop termina infatti troppo rapidamente, la testina legge dei dati senza senso, e quando la CPU tenta di utilizzarli si verificano spettacolari "guru".

È da quando è uscito dalla fabbrica il primo Amiga 1000 che la Commodore mette in guardia contro simili problemi, fornendo anche tutta una serie di soluzioni il cui funzionamento viene garantito su tutti i modelli presenti e futuri. Il fatto che certi programmatori (fortunatamente non tutti) proprio non siano capaci di adattarvisi, mi spinge sovente a paragonarli a dei macachi, paragone che va a tutto vantaggio dei simpatici animali, specialmente quando il programma che ho appena acquistato si rifiuta di funzionare correttamente (qualcuno è mai riuscito a salvare e ricaricare una partita con *Carrier Command*?).

Tra gli applicativi più seri (e costosi) sono pochi quelli che danno realmente dei problemi con l'Amiga 3000, e anche con questi è in genere sufficiente tornare al collaudato *Kickstart 1.3* perché quasi tutti riprendano a funzionare normalmente. È il caso, per esempio di *Excellence!* o di *Sonix 2.0*.

Conclusioni

A conclusione di questo breve sguardo sull'Amiga 3000, si ha come la sensazione di avere solo sfiorato la superficie di qualcosa di potenzialmente molto complesso e sorprendente. A tratti, durante la prova, abbiamo avuto come l'impressione di essere a nostra volta attentamente esaminati dalla stessa macchina e che ci guardasse con uno sguardo del tipo "aspetta solo che abbia i programmi giusti e ti farò vedere di cosa sono capace IO". Quello che per ora manca, in effetti, è proprio il software in grado di gestire adeguatamente tutte le nuove capacità. Come vedremo in un futuro articolo, infatti, i chip ECS hanno ancora molti "assi nella manica" che non sono stati portati alla luce e a volte nemmeno menzionati. Il primo programma a rendere giustizia all'Amiga 3000 dovrebbe essere proprio il tanto atteso *AmigaVision*, del quale vi parleremo diffusamente molto presto.

■

KCS POWER PC BOARD

Finalmente la scheda di compatibilità IBM XT non è più predominio dell'A2000. In soli quindici centimetri un cuore XT anche per il vostro A500

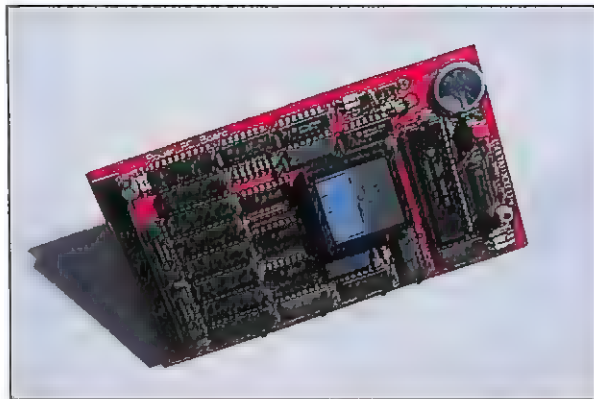
di Nicolò Fontana Rava

Sembra incredibile ma è proprio così: sono finiti i tempi in cui si diceva che l'Amiga 500 era dotato di un connettore per l'espansione di memoria interno. Con l'uscita della scheda Power PC Board infatti, si potrà da ora in avanti parlare di porta per la multicompatibilità.

Questa scheda, prodotta dall'olandese Kolff Computer Supplies e di recente introdotta sul mercato italiano, oltre a contenere una normale espansione di memoria da 512K per l'Amiga, integra al suo interno una vera e propria "scheda madre" di un IBM XT, che, con un minimo di supporto software, è in grado di convertire l'A500 in un PC XT-compatibile al 100%, utilizzando inoltre tutte le porte dell'Amiga, il mouse, i joystick e uno o più drive esterni.

La scheda viene venduta unitamente al sistema operativo Microsoft MS-DOS versione 4.0 completa di manuali, al software di supporto e a un piccolo ma dettagliato manuale per l'installazione della scheda e il suo utilizzo. L'installazione è molto semplice: la scheda si adatta perfettamente allo spazio creato per l'espansione di memoria A501 sul fondo del computer, rimanendo celata e protetta da possibili urti e danneggiamenti. Dopo avere effettuato la semplice operazione il vostro PC è già pronto! È sufficiente riaccendere l'Amiga, inserire i dischi del software di supporto e dopo qualche secondo eccovi davanti allo schermo di un PC dalla tastiera quantomeno singolare...

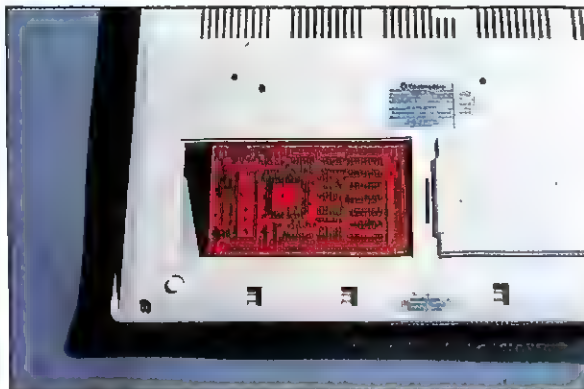
Superata la frenesia del primo momento, peraltro giustificata, è però consigliabile sedersi con calma a leggere attentamente il piccolo manuale d'istruzioni, prima d'iniziare a utilizzare a pieno ritmo il PC. In particolare, è assolutamente necessario effettuare delle copie di backup dei due dischi di supporto dell'emulatore, specie se si intende modificare la configurazione del PC, in quanto non sempre è facile riuscire a tornare alla configurazione originaria; specialmente all'inizio si possono anche combinare pasticci.

Passiamo ora a esaminare più in dettaglio le caratteristiche tecniche della scheda. Per quanto riguarda il suo funzionamento con il sistema Amiga, la scheda mette a disposizione una memoria aggiuntiva di 512K, e un clock hardware supportato da una batteria autonoma della durata di circa dieci anni; tramite il software fornito insieme all'unità è facilmente possibile installare il clock e utilizzare i 512K aggiuntivi come RAM disk. Nella corso della nostra prova non è stato rilevato alcun difetto, incompatibilità o rallentamento nel confronto con la normale espansione prodotta dalla Commodore. In ogni modo è possibile realizzare da soli con facilità un interruttore esterno per disconnettere la scheda senza doverla estrarre, nel caso che qualche programma per Amiga dovesse avere delle difficoltà a funzionare con la scheda collegata. Sul manuale d'istruzioni si trovano indicazioni molto precise su come effettuare questo intervento relativamente semplice.

Ciò che realmente ci interessa è però il PC: per entrare in ambiente MS-DOS è sufficiente inserire il primo disco di supporto all'apparizione della "manina del Workbench" dopo aver acceso il computer. Appare così uno schermo di presentazione della PC Board e dopo qualche secondo... ecco la classica schermata successiva all'accensione di un PC, con intestazione e test della memoria. A questo punto è possibile inserire un qualsiasi programma MS-DOS e utilizzarlo come su un normale PC. La RAM a disposizione dell'utente è di poco superiore ai 700K, in quanto il resto viene utilizzato per controllare la scheda e il sistema operativo. Per quanto riguarda la velocità di elaborazione, il costruttore non fornisce dati, ma dalla prova da noi effettuata i valori si attestano su posizioni simili agli altri PC-compatibili dotati di CPU 8088 a 8 MHz. La prova in questione vede i dati del PC Board confrontati con quelli di computer più potenti e veloci di un XT, e potrà per questo sembrare un po' ingenerosa, ma costituisce in ogni caso un mezzo per avere dei termini di confronto. Il sistema è in grado di supportare fino a un massimo di quattro drive da 3,5" o 5,25"; di emulare due tipi di scheda grafica: MGA (Monochrome Graphics Adapter - Hercules compatibile) che consente una risoluzione di 720 x 348 pixel monocromatica, oppure la diffusa CGA (Colour Graphics Adapter) che utilizza un massimo di 8 colori e una risoluzione massima di 640 x 200, non è prevista invece (e sarebbe forse una pretesa eccessiva) l'emulazione della più recente e potente VGA. È anche possibile utilizzare le porte seriale e parallela dell'Amiga per collegare una stampante o un modem al PC, oltre

alle due porte joy1 e 2 per utilizzare joystick e mouse. Infine c'è da ricordare che alcuni importanti comandi MS-DOS vengono automaticamente assegnati ai tasti funzione dell'Amiga dal programma d'installazione.

Il disco d'installazione della PC Board contiene anche un programma di configurazione del PC che permette di modificarne le caratteristiche nei limiti delle possibilità che abbiamo appena descritto. Accedervi è molto semplice: basta infatti tenere premuto il tasto sinistro del mouse durante il boot del dischetto da sistema AmigaDOS e dopo qualche secondo comparirà il menù principale del programma di configurazione. Naturalmente, il programma è già configurato per rispondere alla maggior parte delle esigenze senza dover accedere a questa fase; la configurazione originaria prevede: modo grafico CGA a colori, un drive interno indirizzato come a: e b:, mouse su joy1 indirizzato come COM2, porta RS-232 dell'Amiga indirizzata come COM1, porta stampante dell'Amiga indirizzata come LPT1. Tutte queste caratteristiche possono essere modificate tramite il programma di configurazione che permette d'intervenire su un elevato numero di parametri. Così per esempio si possono modificare il numero e il tipo di drive, il modo grafico utilizzato e il numero di colori e tutte le porte possono essere assegnate a usi diversi o indirizzate in maniera diversa. Il manuale comunque descrive dettagliatamente ogni possibilità e anche il programma (basato sull'uso di mouse e icone) è decisamente user-friendly e semplice da capire.

Il sistema si integra completamente con l'hardware di cui è già dotato l'Amiga e non è stata riscontrata da parte di chi scrive la necessità di fare complicati interventi o correzioni di nessun genere. La tastiera, i drive, il monitor, la memoria e tutto il resto si adattano con una tale duttilità a operare per i due sistemi che sembra quasi che la Commodore abbia prodotto questo computer con la precisa intenzione di permetterne un giorno la compatibilità con gli "zii" MS-DOS. Anche un eventuale hard disk esterno può facilmente fornire i sui servizi a entrambe i sistemi semplicemente parzializzando il suo contenuto e assegnandone una parte alla memorizzazione di file MS-DOS.

Per quanto riguarda la compatibilità, infine, ancora una volta la PC Board mantiene le sue promesse dimostrandosi affidabile al 100%. Da *PC-Shell* al *WordStar*, da *Lotus* ai videogiochi non siamo riusciti a trovare un programma che avesse delle difficoltà a funzionare. Tutto naturalmente nei limiti delle caratteristiche di questo PC, escludendo quindi i programmi che richiedono la scheda grafica VGA, una memoria superiore a quella disponibile o il supporto di un microprocessore più veloce come l'80286 degli AT

Unico difetto che possiamo trovare a questo PC è quello di non prevedere in qualche modo espandibilità o l'inserimento di nuove schede, ma possiamo considerarlo un difetto? L'aver riprodotto un PC-compatibile che si integra con una macchina sofisticata come l'Amiga, su una scheda di non più di quindici centimetri di lunghezza che va a scomparire in un invisibile cassettino senza occupare spazio, sia un traguardo difficilmente superabile, e più di così... Per questo mi sento tranquillamente di consigliare l'acquisto di questo prodotto ai lettori che avessero esigenze in questo senso. Naturalmente, nessuno deve aspettarsi di guadagnare una potente macchina da ufficio, cosa che nemmeno l'XT è più ormai da tempo; ma certo se si ha bisogno di usufruire anche della vasta biblioteca software disponibile per il mondo MS-DOS questa è sicuramente l'occasione giusta per estendere il fronte della propria compatibilità senza dover cambiare computer.

Il prezzo di 749.000 lire, anche se non bassissimo, è abbastanza adeguato al prodotto: in pratica si tratta di un vero e proprio computer su una minischeda. Complimenti dunque ai tecnici della KCS e a chi, avendo capito i pregi e anche i limiti di questo piccolo gioiello, non resterà sicuramente deluso dall'acquisto della scheda PC Board.

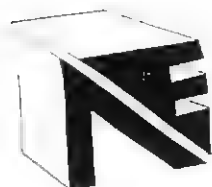
E per chi volesse disporre di un AT invece che di un XT la KCS ha appena prodotto anche una scheda – questa volta esterna – di compatibilità AT. Dovrebbe essere già disponibile nel momento in cui leggete questo articolo. ■

La Power PC Board è disponibile presso:

Newel srl
(Power PC Board, L. 749.000 Iva inclusa)
Via Mac Mahon, 75
20155 Milano
(Tel. 02/323492 - 93580086)

I RISULTATI DEL TEST

	PC Board 8Mhz	6 MHz	*PC 40-III* 8MHz	12 Mhz	*IBM AT* 6 MHz
TEST GLOBALI					
Dhrystone					
senza registri	**847**	1086	1562	2272	1086
Sieve					
con 100 iterazioni	**99**	69	51	33	69
Big sieve					
con 10 iterazioni	**281**	193	143	95	n/d
TEST SPECIFICI					
Loop di conteggio					
Long 10000	**102**	75	54	36	75
Integer 10000	**56**	38	29	19	38
Register 10000	**17**	16	12	7	15
Stampa di 1000 stringhe					
Modo testo MDA	**22**	21	18	12	14
Modo testo CGA	**30**	24	18	12	n/d
Spostamento di blocchi					
Copia stringhe (70 char) 20000	**42**	33	25	[illegible]	[illegible]
Copia strutture (160 byte) 20000	**26**	[illegible]	15	[illegible]	[illegible]
10000 chiamate a funzioni					
Senza parametri	**23**	8	[illegible]	[illegible]	18
Con tre parametri numerici	**45**	32	24	[illegible]	32

Nota: i test offrono risultati espressi in secondi. Numeri inferiori corrispondono a prestazioni migliori. L'unica eccezione è Dhrystone: maggiori Dhrystone/secondo corrispondono a maggiori prestazioni.

NEWEL

Via Mac Mahon, 75 - 20155 MILANO
negozio: 323492 (solo Milano)
AL MATTINO DAL MARTEDÌ
AL VENERDÌ 93580086 (solo ordini)
Solo per urgenze: 0337/277389
Fino alle ore 18: 33000036 (ordini certi)
3270226 (ordini con servizio tecnico)

DRIVE INTERNI/ESTERNI PER TUTTI GLI AMIGA ALTA QUALITÀ

GARANZIA 12 MESI

• DRIVE INTERNO PER AMIGA 500	L. 189.000
• DRIVE ESTERNO PER AMIGA 500/1000 PASS. + DISCONNECT	L. 179.000
• DRIVE INTERNO PER AMIGA 2000 (COMPLETO)	L. 169.000
• DRIVE ESTERNO PER AMIGA 500/1000/2000 DA 1,52 MB!	L. 299.000
• DRIVE ESTERNO PER AMIGA 500/1000/2000 DA 5"1/4	L. 299.000

C'È BULK E BULK. NOI TI OFFRIAMO DISCHETTI DI QUALITÀ 3 1/2 DS-DD CERTIFICATI UNO AD UNO.
CONF. DA 50 PZ. L. 1.200 CAD.
CONF. DA 100 PZ. L. 1.000 CAD.
CONF. DA 500 PZ. L. 900 CAD.
TUTTI I PREZZI IVA COMPRESA

MODEM-PROFESSIONALI
Supramodem
300/1200/2400 band.
L. 339.000

Ora disponibile anche interno per A2000 a sole
L. 299.000

Novità disponibili modem HST 19200 HIGHSPEED, correzione d'errore **L. telef.**

NOVITÀ ASSOLUTA

• novità per chi già possiede una scheda xt per a2000

• turbo-xt l. 179.000
raddoppia la velocità del clock, indispensabile.

• xt-ram 768k l. 279.000
porta la memoria della vostra scheda xt a 768 su piastra madre.

• 386-sx power card l. 1.290.000
trasforma la vostra scheda xt in una at-386sx a 16 mhz con 16kbyte di cache-memory, aumenta la velocità da xt fino a 12 volte, predisposto per coprocessore 387-sx. (straordinaria!!!)

• trackdisplay per amiga 2000 l. 169.000
utile novità segnalatore di tracce per due drive a due hard, solo per amiga 2000, semplice installazione.

• emulatori
finalmente disponibile l'ultima versione hardware dell'atari st emulator per amiga 500/2000 per una quasi totale compatibilità al 99%. l. 399.000

• a-max II disponibile nuova versione, ed upgrade per chi possiede già la prima

• alimentatore professionale per amiga 500 (disponibile) l. 125.000

CON L'ARRIVO DELL'AUTUNNO ALLA NEWEL ARRIVANO FANTASTICHE NOVITÀ PER AMIGA!!!

un nuovo digitalizzatore a colori professionale per amiga!
un nuovo genlock semiprofessionale con effetti video ad un prezzo eccezionale!!!
e decine di nuovi articoli richiedi il nuovo catalogo newel disponibile da metà ottobre, vi attendono fantastiche novità e non solo...?!?!?!!

• ESPANSIONI DI MEMORIA PER AMIGA 500, 1000 E 2000

Costruite con i migliori materiali, le nuove espansioni di memoria dell'ultima generazione usano i nuovissimi chip da 1 mbit che sono notevolmente più veloci, autoconfiguranti, slim line, e con 1 anno di garanzia! Aggiungono rispettivamente alla memoria base da 512k a 8 mb

• 512k interna per amiga 500	l. 119.000
• 512k interna per amiga 500 + clock	l. 139.000
• 1,5 mb interna per amiga 500 + clock	l. 350.000
• 2 mb interna per amiga 500 + clock	l. 450.000
• 4mb interna per amiga 500 + clock	l. telef.
• 2 mb esterna espandibile a 8 mb per amiga 1000	l. 599.000
• 2 mb interna per amiga 2000 espandibile a 8 mb	l. 650.000
• 8 mb interna per amiga 2000 con 8 mb ram!	l. 1.500.000

"MIDI"
digitalizzatore audio-stereo pro-sound designer gold v.2.
l. 175.000

INTERFACCIA MIDI PROFESSIONALE
2 in - truh - 3 out l. 79.000

Nuovo AGNUS 8372-A
L. 138.000
completo di istruzioni

Tutte le ultime novità software originali importazione diretta!!!

Tutti i nostri prodotti sono coperti da garanzia di 12 mesi.

Tutti i prezzi sono IVA 19% compresa

NEWEL

le novità "newel" autunno 1990

PAL GENLOCK 2.0

nuovo pal genlock semi-professionale con regolazioni esterne per risultati ancora migliori, disponibili varie versioni

L. 399.000

SCANLOCK-GENLOCK NOVITÀ

eccovi uno dei migliori genlock per tutti gli amiga funziona su a500/1000/2000/2500. il supergenlock è il primo genlock con qualità broadcast ed uscita in super-vhs, m2 betacam, luma/croma o y/c, banda passante fino a 5,5 mhz. regolazione di fase, è possibile ad esempio cambiare da pal, s-vhs a rgb senza spostare nessun cavo anche in fase di editing. inoltre key out per croma key, key reverse, normal key, questo e molto di più puoi ottenere con il nuovo super-scanlock.

OFFERTA LANCIO

L. 1.990.000

AMIGA SELECTOR
(novità esclusiva NEWEL)

ora che inizia l'abbondare delle periferiche ed accessori per amiga, nasce il problema che molti si collegano alla porta parallela, che è quasi sempre già impegnata con la stampante, per questo motivo la newel ha creato questo nuovo dispositivo che permette di collegare fino a quattro accessori/interfacce come ad esempio: digitalizzatori video (easy/view, digiview, videon) digitalizzatori audi (easysound, prosound .) ed altri accessori come amigafax e naturalmente le stampanti amiga selector è dotato di un commutatore che seleziona l'interfaccia desiderata amiga selector è molto utile come prolunga di circa 40 cm della porta parallela, molto utile ad esempio per il videon II che è corredato da un cavo molto corto, appena entrerete in possesso di questo utilissimo accessorio vi renderete conto che è indispensabile!

L. 99.000

PENNA OTTICA AMIGA

penna ottica amatoriale per tutti gli amiga, completa di software di gestione, funziona in emulazione mouse ed è compatibile con la maggior parte del software in commercio, corredata di manuale di istruzioni interno in italiano

L. 49.000

disponibile anche nuova versione professionale!

AMIGA TELEVIDEO

L. 199.000

INTERESSANTISSIMA INTERFACCIA PERMETTEDI RICEVERE IL SEGNALE TELEVIDEO (DALLE PRINCIPALI TV COLLEGATE COME: RAI, TELELOMBARDIA ECC. ECC.) DI SEMPLICE INSTALLAZIONE PERMETTE OLTRE LA VISUALIZZAZIONE A VIDEO LA POSSIBILITA DI SALVARE SU DISCO E DI STAMPARE DIRETTAMENTE. UTILISSIMO. SOLO PER AMIGA 500/2000

SYNCRO EXPERT - VERSIONE III - NOVITÀ **ECCEZIONALE**
SERVE SOLO PER COPIE DI SICUREZZA AD USO PERSONALE

se hai l'amiga ed un secondo drive, questo prodotto non puoi assolutamente perderlo. qualsiasi programma, originale e non, potrai finalmente duplicarlo per le tue copie di sicurezza. copia qualsiasi formato, anche ms/dos 3.5", mac, atari, ecc. su dischi 3.5" fino a 80 tracce 2 facciate. semplicissimo da usare, non necessita di alcuna conoscenza. la sua forza è un chip custom lsi che si sovrappone al microprocessore copiando qualsiasi disco.

lit. 85.000 il solo copiatore. lit. 249.000 il copiatore più un secondo drive con sconnettore slim - meccanica citizen.

VIDEON III⁰

PER AMIGA 500/2000 PRESTAZIONI MAI VISTE, DIGITALIZZATORE PROFESSIONALE DIRETTAMENTE A COLORI

IL PREZZO??? ECCEZIONALE. TELEFONA!!!

NOVITÀ ASSOLUTA PC-BOARD PER AMIGA 500 NEW "PAL" VERSION L.749.000

Incredibile. Con meno di 750.000 lire puoi avere nel tuo amiga due computer. Puoi renderlo compatibile ibm in tutto e per tutto, compatibile con il mouse, joystick, drive interno ed esterno (anche 5" 1/4), parallela e seriale in linea. 1 mb di ram + orologio visto anche sotto amigados, msdos ultima versione + utility in omaggio. La scheda completa e semplicissima da installare, si infila nello slot dell'espansione di memoria!!! Nuova versione pal!!!

ACTION REPLAY NOVITÀ

la prima cartuccia rivoluzionaria multifunzioni per amiga 500/1000 con opzioni di freeze permette di sproteggere la maggior parte dei programmi in commercio (consentendoti di creare giochi di sicurezza per uso personale, inoltre permette di creare giochi trainer. (vite infinite ecc.), permette di bloccare un gioco in qualsiasi momento dal medesimo posto salva una qualsiasi videata (disegno, testo) su disco, consentendoti una facile hardcopy anche su stampante. funzione moviola (rallenta programmi e giochi). potente virus-detector, sprinteeditor, oltre che ad un monitor straordinario per il linguaggio macchina questo e molto, molto di più. ti aspetta in amiga action replay'!!
il tutto ad un prezzo eccezionale!
versione originale con manuale in italiano

L. 169.000

"AMIGA ADVANCED UTILITY KIT" NOVITÀ

upgrade del software, notevolmente potenziato (professional trainer, sound tracker-ripper, nuovo convertitore di files compatattore) più numerose altre utility.
n.b. tutti i bug della precedente versione sono stati rimossi.
il tutto con un nuovo manuale complementare in italiano di oltre 20 pagine.
il costo? l. 29.000 originale e per chi lo acquista con la cartuccia solo

L. 15.000

MOUSE SELECTOR NOVITÀ

novità che permette di connettere alla porta 1 (mouse) dell amiga contemporaneamente il mouse ed il joystick e selezionare a piacere senza staccare ogni volta i cavi. utilissimo!!!

L. 29.000

A520 EXPANDER

(novità assoluta esclusiva newel)
finalmente una soluzione per lo scomodissimo modulatore tv (a520), con questo cavo speciale puoi tenere il modulatore a 20/30 cm dall'amiga senza più intralciare. utilissimo!!!

L. 29.000

DISK-PROTECTOR NOVITÀ

spesso per una piccola distrazione rischiate di danneggiare irrimediabilmente i vostri dischetti da virus, o semplicemente formattandoli ecc con questo dispositivo hardware potete impedire che si possa scrivere sul disco indipendentemente dalla finestrella del disco. molto utile!

L. 29.000

PROLUNGA DRIVE

prolunga schermata per drive aggiuntivi

L. 39.000

ALIMENTATORE STABILIZZATO PER AMIGA 500

L. 125.000

220V PROTECTION

spesso molti danni del vostro computer & periferiche (stampanti, monitor ecc.) sono dovuti a sbalzi di corrente, disturbi ed altro. con questa nuova presa multipla appositamente studiata per computer potete collegare fino a 8 apparecchi/periferiche proteggendoli. dotato di interruttore generale. utilissimo (funziona con tutti i computer)

L. 69.000

fish-disk aggiornati al n° 350 tutti!

DISPONIBILE A MAGAZZINO L'INTERA GAMMA PROFESSIONALE AMIGA 3000

AMOS: UN NUOVO LINGUAGGIO DI PROGRAMMAZIONE

L'analisi di un interprete per l'Amiga dalle caratteristiche a dir poco singolari. Progettato per la costruzione di programmi che fanno un elevato uso di grafica, suoni e animazioni, mette a portata di... Basic tutta la potenza dell'Amiga

di Avelino De Sabbata

Tutti conoscono le caratteristiche che rendono l'Amiga un computer unico e non è certo lo scopo di questo articolo dimostrarlo. Per la stesura di programmi che facciano un uso approfondito delle opportunità che offre l'hardware dell'Amiga però, fino a oggi era necessario rivolgersi a linguaggi tipicamente vicini alla macchina quali il C o addirittura l'Assembly. Diciamo "fino a oggi", perché le cose stanno cambiando, e, come abbiamo visto sul numero 4/90 di *Commodore Gazette* dove ci siamo occupati di *GFA BASIC*, anche i Basic disponibili per l'Amiga stanno moltiplicandosi ed evolvendo, addirittura specializzandosi per la produzione di particolari applicazioni come nel caso del Basic oggetto di questo articolo.

AMOS originariamente avrebbe dovuto essere chiamato THE GAME CREATOR, ma viste le ampie possibilità di utilizzo del linguaggio la parola GAME è stata eliminata, perché troppo riduttiva rispetto alle prestazioni offerte.

AMOS è stato scritto da François Lionet, un valentissimo programmatore, autore tra l'altro dello STOS, il GAME CREATOR molto diffuso tra gli utenti Atari ST, dal quale è stata eseguita la trascrizione per Amiga. François, da utente ed esperto programmatore Atari, dopo l'impatto con l'Amiga si è talmente innamorato di questo computer che ha deciso di rinnegare l'ST per dedicarsi unicamente all'Amiga.

Il pacchetto viene distribuito in una confezione di solido cartone, dalle dimensioni contenute, ma dal cui peso si deduce che il ridotto spazio è completamente utilizzato. All'interno della scatola troviamo infatti tre dischi, *Program, Extras* e *Data*, accompagnati da due manuali, *User Guide* e *Extras Manual*, il numero zero di una newsletter redatta dall'*AMOS* User Club inglese sulla quale sono riportante curiosità, novità e trucchi riguardanti la programmazione in AMOS, un fascicoletto che riporta un indice ragionato e un elenco dei numerosi messaggi di errore completo di una breve spiegazione per ognuno, un foglio che dà una velocissima spruzzata di *AMOS* all'utente che vuole subito testarne il polso, e infine l'immancabile cartolina di registrazione. Il tutto naturalmente scritto in lingua inglese.

Nei tre dischi in dotazione troviamo un'innumerevole quantità di directory e file. In particolare, sul *Program Disc*, notiamo la directory AMOS_SYSTEM la quale contiene alcuni file di configurazione dell'ambiente *AMOS* che ci permettono di ridefinire completamente l'ambiente di programmazione non solo nelle dimensioni e nei colori dello schermo di lavoro, ma ben più in profondità, offrendoci la possibilità di ridefinire la tastiera, la fonte da utilizzare, i vari puntatori e pattern previsti dall'interprete, l'icona... Tramite un opportuno programma possiamo infatti ridefinire anche tutti i messaggi interni di *AMOS*, che assieme a tutte le altre variazioni possono essere salvati nel file AMOS.Env.

MANUAL, la seconda directory che attira la nostra attenzione, si rivela una notevole fonte d'informazioni ed esempi di programmazione: divisi per capitoli troviamo un centinaio di programmi sull'uso del linguaggio ai quali viene costantemente fatto riferimento nel corso del manuale. Nella directory Help una serie di brevi file di testo danno alcune indicazioni sull'uso di alcuni programmi e utility forniti nei tre dischi.

Alcune utility permettono infine di disegnare sprite e oggetti, definire parte di schermate grafiche in sprite e bob, convertire una determinata fonte carattere per essere usata dall'editor di *AMOS*, ridefinire la tastiera, convertire moduli musicali creati con *Soundtracker*, *Sonix* o *GMC* in formato adatto a essere usato in *AMOS*, creare banchi di suoni campionati, installare *AMOS* su disco rigido e svariate altre operazioni.

Il disco *Extras* contiene alcune demo, tra cui un simpatico *Scrolling_text_demo.AMOS*, che crea un testo scritto in caratteri molto grandi e variopinti fatto scorrere orizzontalmente sullo schermo con una fluidità, e una velocità, ottenibili prima d'ora solo in Assembly. Il programma *Sprite600.AMOS* ci offre una frizzante panoramica di ciò che si può ottenere con sprite e bob, in una interminabile sequenza di animazio-

ni, per le quali vengono usati ben 600 oggetti differenti.

Alcuni programmi di utilità estremamente efficaci mettono a disposizione un editor/monitor per costruire e verificare complesse animazioni, un editor con cui costruire esternamente al programma sofisticati menu, per i quali *AMOS* prevede estreme raffinatezze. Un MAP_Editor permette di costruire schermate di fondo per quei giochi che prevedono un'innumerevole quantità di quadri ma che possono essere costruiti con relativamente poche icone di base.

RAMOS_Install.AMOS permette di creare dischi contenenti programmi *AMOS*, che possono funzionare indipendentemente dall'interprete grazie al modulo di runtime RAMOS, e pertanto distribuibili senza infrangere copyright. A questo riguardo segnaliamo la prevista disponibilità per dicembre dell'anno in corso del compilatore, al quale è attualmente impegnato lo stesso François Lionet. Grazie alla compilazione, a detta della Mandarin Software, sarà possibile ottenere programmi eseguibili molto compatti, e perfino più veloci (circa il doppio), dei già velocissimi programmi *AMOS*.

Quello che si può dire da una prima osservazione dei due manuali forniti, è che per essere fatti meglio, potrebbero solo essere in italiano. Infatti nella *User Guide*, che conta circa trecento pagine rilegate a spirale e stampate fitte fitte, troviamo la spiegazione di tutti i comandi divisi in 22 capitoli (per la maggior parte dei quali viene riportato un breve esempio d'uso), raggruppati per funzione logica. Il bello è che la trattazione di ogni comando non è fine a se stessa, ma segue un filo conduttore che capitolo per capitolo, ci introduce gradualmente all'apprendimento del linguaggio e di alcune tecniche assolutamente inedite. Il che rende piacevole la lettura e svela con semplicità tutta la potenza delle numerosissime istruzioni, quasi 500!

L'*Extras Manual*, un fascicoletto di 36 pagine, riporta le indicazioni per la creazione di un disco distribuibile utilizzando il modulo RAMOS, e in modo abbastanza completo le istruzioni per le più interessanti utility e per i programmi dimostrativi forniti.

Sfogliando velocemente la *User Guide* vedremo ora di capire assieme quali sono le caratteristiche di questo linguaggio, che fa un uso rivoluzionario di schermi e finestre, non adeguandosi a ciò che chiunque si aspetterebbe di trovare inalterato nell'Amiga: lo standard imposto da Intuition. In questa escursione ci soffermeremo a illustrare quelle caratteristiche che fanno di *AMOS* un Basic così particolare, ponendo l'accento su ciò che (in Basic) costituisce un'assoluta novità.

Già fin dal secondo capitolo, dove vengono indicate le procedure d'installazione, è possibile avere un assaggio della facilità di lavoro: un breve tutorial ci spiega le operazioni principali. Caricare, salvare o cancellare un programma, caricare un modulo musicale e ascoltarlo, caricare un banco di sprite, visualizzarli...

Nel capitolo successivo, un secondo tutorial ci aiuta a prendere confidenza con l'editor e con le numerose funzioni ottenibili sia con sequenze di tasti, sia da mouse, sia con i tasti funzione. L'editor di *AMOS* si presenta con due barre di scorrimento per il testo e una zona superiore nella quale vi sono dieci gadget che riportano la funzione assegnata ai rispettivi tasti funzione. Questi gadget vengono evidenziati quando il puntatore vi è posizionato in corrispondenza, e l'attivazione della funzione si ha quando il gadget viene selezionato col mouse. Con la pressione di un tasto di controllo (CONTROL, SHIFT, ALTernate, e Amiga destro e sinistro) si ottiene la visualizzazione nei gadget delle rispettive funzioni assegnate, che possono essere selezionate con la pressione del corrispondente tasto funzione o con la selezione tramite mouse.

L'editor di AMOS

Durante la stesura dei programmi è l'editor stesso che provvede a convertire tutti i nomi di variabili, etichette e procedure in maiuscolo, mentre i nomi dei comandi (che possono anche essere composti di 2 o 3 parole come per esempio "Get Sprite Palette"), vengono trasformati in minuscolo, con la sola iniziale di ogni parola in maiuscolo.

Gran parte delle funzioni dei menu si possono ottenere anche con la pressione di un tasto alfabetico (al posto del tasto funzione), che generalmente è il carattere iniziale del comando assegnato. Sotto alle due righe di gadget, c'è una riga di stato dove vengono costantemente riportate alcune informazioni quali il modo di scrittura attivo (Insert/Overwrite), lo stato di CAPS LOCK, la posizione del cursore nel testo, l'ampiezza della memoria a disposizione per il testo, la quantità di memoria disponibile nel sistema (divisa in CHIP e FAST) e il nome del programma su cui si sta lavorando.

Stando così le cose, la visualizzazione del listato è ristretta a una finestra di 22 righe lunghe 78 caratteri. Un po' troppo poco, anche se lo scroll del testo è molto veloce. *C1 TEXT*, che sto utilizzando in questo momento, mi permette la visualizzazione di ben trenta righe di testo, utilizzando un sistema di scroll altrettanto valido! E siamo così arrivati al primo appunto da fare ad *AMOS*:

un editor molto sofisticato e veloce, che però a causa della particolare gestione dei menu a gadget (avrei preferito menu a tendina), limita lo spazio riservato all'area di lavoro, anche se tramite il programma di configurazione è possibile ovviare parzialmente al problema: impostando opportunamente alcuni parametri si riesce a portare l'altezza dell'area di edit fino a 25 righe. Da notare che non è previsto l'uso di uno schermo interlace.

Una veloce carellata su alcune delle numerose opzioni ci darà l'opportunità di scoprire alcune importanti particolarità di *AMOS*. Sulla prima (RUN) non c'è molto da dire, se non che a ogni partenza *AMOS* controlla la correttezza formale del programma, la chiusura dei cicli, delle procedure, l'esatta sintassi..., cosa che del resto è possibile anche con la seconda opzione (F2). Grazie al riconoscimento dei cicli di programma da parte dell'editor di *AMOS*, è possibile con F3 far eseguire l'indentazione automatica del listato, che, a differenza di *GFA BASIC* (recensito nel numero precedente), non viene effettuata automaticamente a ogni linea introdotta.

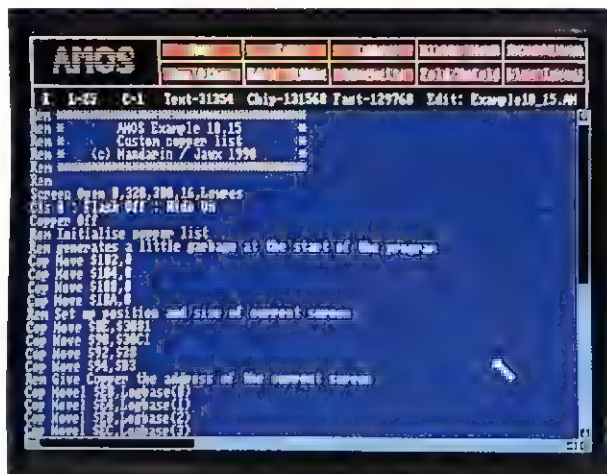
F4 non fa altro che attivare il BLOCKS MENU, e in pratica equivale alla pressione del tasto CONTROL. Lo stesso vale per F5 che attiva invece il SEARCH MENU, ottenibile anche con la pressione del tasto ALT. L'unica differenza in questi due casi è che se i menu (BLOCK o SEARCH) vengono attivati con la pressione del tasto funzione rimangono attivati fino alla scelta di una loro funzione, mentre se l'attivazione è ottenuta con il relativo tasto di controllo (CTRL o ALT), questa permane fino a che persiste l'azione sul tasto di controllo relativo. E fino a questo punto niente di granché interessante, ma l'esame dell'opzione abbinata a F6 ci porta a scoprire che *AMOS* offre la possibilità di avere in memoria più programmi contemporaneamente. Infatti, con la pressione del tasto F6 si ottiene la presentazione di un ottimo e veloce selettore di file dal quale scegliere quale programma eseguire. Memoria permettendo, è possibile avere "on-line" fino a sedici programmi contemporaneamente, fra i quali però uno solo per volta potrà essere editato (F7) e mandato in esecuzione. *AMOS* esegue una distinzione tra programma principale, e programmi "accessori", ma la distinzione è puramente formale, in quanto qualsiasi programma può venire caricato come programma principale, oppure come programma accessorio. La distinzione acquista particolare valore nel caso di alcune utility che hanno effettivamente una funzione ausiliaria al programma principale, che possono per esempio essere utilizzate nella fase di stesura di un programma. Tra il software fornito in dotazione, ne possiamo citare alcune che rendono bene l'idea: *Sprite Grabber, Sprite Editor, Map Editor, AMAL EDitor, Menu Editor* (programmi che hanno in comune la caratteristica di poter accedere ai dati utilizzati dal programma principale).

Si noti però che l'accesso a più programmi non è una prerogativa della fase di programmazione, ma può essere prevista anche in fase di runtime, passando così dall'esecuzione di un programma all'altro. In questi casi l'ambiente di ogni programma rimane completamente indipendente, e ognuno conserva le proprie variabili separate da quelle degli altri. Per questo scopo *AMOS* prevede alcune istruzioni: BGRAB permette di "rubare" i banchi di memoria occupati dai dati utilizzati da un altro programma, PRUN congela il programma corrente e manda in esecuzione un altro programma in memoria, le funzioni PRG FIRST$ e PRG NEXT$ permettono di conoscere il nome dei programmi rispettivamente precedente e successivo al programma corrente nella lista dei programmi in memoria, mentre PSEL$ permette di eseguire un programma selezionandolo dal selettore di file standard *AMOS*.

Con il tasto funzione F8, notiamo un'altra importante caratteristica: il collassamento delle procedure, che come abbiamo avuto modo di osservare nel caso del *GFA*, rende particolarmente agevole l'esame e l'elaborazione di un listato. Una procedura può essere richiusa su se stessa mostrando nel listato solo la sua testa. Questi brani di programma, diventano così inaccessibili sia alla vista che alle modifiche. Infatti, impartendo l'ordine di sostituzione di una determinata ricorrenza, la sostituzione stessa potrà avere esito positivo all'interno della procedura solo nel caso in cui questa sia aperta, che il proprio codice sia cioè visibile. Il tasto F8 premuto quindi in corrispondenza di una qualsiasi istruzione di una procedura ha l'effetto di collassarla, premuto nuovamente, quando il cursore si trova sull'unica linea visibile della procedura richiusa, la farà riespandere. La coppia di tasti funzione ALT- F7/F8, ha lo stesso effetto invece su tutte le procedure del listato, e in questo caso è ininfluente la posizione del cursore. Ciò che in alcuni casi risulta scomodo nelle operazioni di Fold-Unfold, è il fatto che il sistema, all'atto dell'apertura o della chiusura di una o di tutte le procedure, effettua un test di tutto il programma: questa operazione nei programmi brevi o di media lunghezza viene eseguita quasi istantaneamente, ma per programmi di una certa complessità il pur breve tempo da attendere inoperosi davanti al monitor risulta fastidioso; avremmo preferito che per l'apertura o la chiusura di un'unica procedura, *AMOS* pensasse a verificare la correttezza della sola sezione di programma interessata.

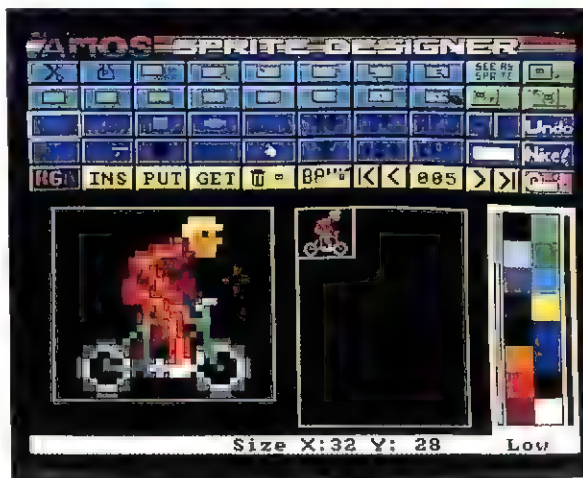
L'ultimo tasto funzione, F10, ci permette di ottenere l'inserimento di una riga vuota nel listato. Lo stesso effetto si ottiene con la pressione di CTRL I, oppure con l'azione sul tasto RETURN quando il cursore si trova sul primo carattere di una riga.

Grazie alla serie di funzioni ottenibili con la pressione contemporanea del tasto SHIFT, è possibile caricare o salvare un programma in formato tokenizzato ed effettuare un MERGE con codice sia in AMOS sia in ASCII, caricare programmi accessori e altri programmi

AMOS, cancellare un programma ausiliario oppure tutti i programmi e i dati in memoria, oppure uscire da *AMOS*.

La serie di opzioni ottenibili con la pressione contemporanea del tasto CTRL, permette di definire un blocco; notiamo con soddisfazione che lo stesso si può ottenere anche da mouse. Come nel caso del *GFA* anche *AMOS* non permette di definire come blocco solo parte di una linea, anche se in questo caso c'è maggiore flessibilità in quanto è possibile spezzare in due parti un'unica linea di programma, o fondere due linee in una sola. Sempre nel menu BLOCKS troviamo la possibilità di rimuovere un blocco, muovere o copiare il blocco definito nella posizione corrente del cursore, cancellare la definizione attuale che si ottiene anche con un semplice click del pulsante destro del mouse, stampare e salvare il blocco definito, sia in ASCII sia in formato AMOS, e copiare il blocco corrente in un particolare buffer in modo da essere prelevato da un altro programma in memoria; questo rende possibile lo scambio di brani tra i vari programmi. Da osservare che è possibile la definizione di tutto il programma come blocco con la sequenza CTRL A, il che risulta utile per salvare tutto il programma in formato ASCII, non essendo prevista la funzione specifica. Inoltre, a causa di un grossolano bug, l'istruzione LLIST che dovrebbe produrre la stampa del listato non viene riconosciuta; fortunatamente ciò si può aggirare con facilità definendo tutto il programma come blocco e mandandolo in stampa: CTRL A + CTRL-F10.

Dell'ultima serie di opzioni che si ottengono con il tasto ALT, le prime sei permettono una agevole ricerca e sostituzione di una specificata ricorrenza nel testo corrente in entrambi i sensi. Le ultime opzioni di questo menu hanno invece una funzione generale: con Set Buffer si può stabilire la quantità di memoria da destinare al programma corrente, e con Set Tabs si ridefiniscono le posizioni di tabulazione, che vengono utilizzate nell'indentazione automatica.

Entriamo in modo diretto

Dopo aver mandato in esecuzione il programma, la pressione del tasto ESCAPE provoca lo scorrimento dei due schermi che compongono l'editor (uno con menu e linea di stato e uno con l'area lavoro), che con un simpatico effetto rendono visibili altri due schermi. Il primo è lo schermo di output che per default ha una dimensione di 320 x 200 punti in bassa risoluzione con sedici colori, mentre una seconda view in alta risoluzione (lo schermo di input) alta una decina di linee di testo viene visualizzata nella parte bassa del monitor. Su questo secondo schermo, le cui dimensioni e posizione possono essere modificate, lampeggia un cursore in attesa di comandi. L'effetto dei comandi immessi in modo diretto è visualizzabile sullo schermo in bassa risoluzione. Dal modo diretto, una seconda pressione del tasto ESCape, oppure l'immissione del comando EDIT, ci riportano agli schermi dell'editor.

Per quanto concerne l'edit di testi molto lunghi c'è da osservare che è possibile memorizzare fino a dieci punti particolari, dove sarà in seguito facile portare immediatamente il cursore con la semplice pressione di due tasti. La ricerca delle procedure poi è molto facilitata: infatti, la pressione di ALT più un tasto cursore UP/DOWN, porta il cursore immediatamente all'inizio della procedura precedente o seguente.

Il manuale prosegue, dando alcune indicazioni sul modo di risparmiare un po' di memoria. L'interprete *AMOS* infatti necessita di una notevole quantità di RAM per fuzionare, e i programmatori intenzionati a usare *AMOS* con un Amiga 500 inespanso, dovranno prestare particolare attenzione a questa paginetta, tenuto conto del fatto anche che in alcuni casi è veramente molto utile la possibilità di avere più programmi in memoria. Allo scopo di aumentare i sempre preziosi "bytes free", *AMOS* riserva anche due istruzioni: Close Workbench e Close Editor. La condizione necessaria perché la prima possa avere successo è data dal fatto che *AMOS* non sia mandato in esecuzione in background e che nessun altro programma stia funzionando. Al RUN, *AMOS* per default tenta di chiudere lo schermo del *WB*, e se questo avviene non sarà più possibile uscire da *AMOS*, se non con un reset totale. Questo può essere evitato modificando il file AMOS.Env tramite il citato programma *Config.AMOS*, oppure semplicemente con l'uso in background di un breve programmino che non usi troppi cicli della CPU (per esempio *VirusX*). Da osservare che *AMOS* supporta il multitasking, però, evidentemente imposta la propria priorità a un valore talmente elevato che le operazioni in corso negli altri processi ne vengono altamente penalizzate.

Con *AMOS* in background è possibile osservare un fatto curioso: i tasti normalmente utilizzati per portare lo schermo del *WB* in primo piano non sortiscono alcun effetto. *AMOS* sembra così l'unico processo in memoria. Fortunosamente, ho però scoperto che la pressione contemporanea di Amiga sinistro + "A" (non documentata) ci riporta al *WB*, e allo stesso tempo sembra far *sparire* il processo *AMOS*. Come d'incanto però, una nuova pressione dei tasti sopra citati, ci fa riapparire *AMOS* nelle condizioni in cui lo avevamo lasciato. L'arcano è però presto svelato: *AMOS* supporta gli schermi logici, che esistono a tutti gli effetti in memoria, ma che non sono visibili!

Con la lettura del quarto capitolo dedicato ai principi di base scopriamo la semplicità della gestione delle variabili in *AMOS*, che ne riconosce solo tre tipi diversi: interi, virgola mobile e stringhe. Gli interi che utilizzano 4 byte per la loro memorizzazione e possono contenere valori da −147.483.648 a +147.483.648, sono il tipo utilizzato per default, in quanto vengono gestiti con velocità notevolmente maggiore delle variabili in virgola

mobile. A questo secondo tipo di variabili possono essere assegnati valori compresi nel range 1E-14 e 1E+15, e l'identificatore che le distingue è il carattere #. Molto flessibile e veloce è la gestione delle variabili stringa, individuate dal classico "$", le quali possono assumere la rispettabile lunghezza di ben 65.500 caratteri. Oltre alle classiche operazioni, segnaliamo la possibilità di eseguire la sottrazione tra stringhe, così che per esempio l'operazione "AMOS BASIC"–"BASIC" darà per risultato "AMOS ", mentre tra le funzioni troviamo l'inconsueta FLIP$, che restituisce il "rovescio" della stringa argomento: per esempio, FLIP$("AMOS") darà per risultato "SOMA". Tra le operazioni con gli array ci preme sottolineare SORT che riordina in modo ascendente ogni tipo di matrice, e MATCH, che cerca una determinata ricorrenza in un array che dev'essere preventivamente riordinato.

Lo stesso capitolo prevede un'ampia trattazione di procedure, parametri, numeri di linea (che non sono indispensabili, ma sono supportati unicamente per la compatibilità con lo STOS) ed etichette. Per quanto riguarda le procedure i parametri possono essere passati come costanti, variabili oppure espressioni. Le variabili all'interno delle procedure sono locali e vengono cancellate all'uscita della procedura. Può essere utilizzato il comando SHARED all'interno di una procedura per individuare quelle variabili definite nel programma principale, che debbono essere visibili anche al suo interno, mentre con il comando GLOBAL, solitamente all'inizio del programma (e naturalmente fuori dalle procedure), vengono definite quelle variabili che saranno condivise da tutte le procedure. È comunque possibile ricevere dei valori di ritorno anche da quelle procedure che utilizzano solo variabili locali riportando la variabile da ritornare dopo la parola END PROC, che può così essere intercettata dal programma grazie alla parola PARAM. Notiamo il comando POP PROC che permette di uscire da una procedura al verificarsi di una determinata condizione.

Una parte estremamente interessante di questo capitolo è riservata alla gestione della memoria: la manipolazione di una notevole quantità di dati non omogenei tra loro, come sprite e moduli musicali, è estremamente facilitata dal fatto che è previsto un particolare formato per alcuni tipi di dati, e che sono messe a disposizione del programmatore fino a 15 zone di memoria (banchi). Per questi banchi esiste un comodo set d'istruzioni con le quali eseguire l'allocazione, che per alcuni tipi di dati è automatica, la cancellazione, il caricamento e il salvataggio verso la memoria di massa, sia in formato AMOS, sia in formato binario. È possibile infine salvare ogni banco di memoria separatamente uno dall'altro, oppure salvare il programma completo in un unico file, contenente al suo interno tutti i dati. La differenza tra il formato AMOS e il formato binario sta nel fatto che i file creati con il primo, contengono una particolare header che definisce il tipo di dati del file, così che al prossimo LOAD, i dati potranno essere automaticamente indirizzati verso i banchi di memoria da cui erano stati prelevati da LOAD. Il comando LIST BANK impartito in modo diretto produce l'elenco dei banchi occupati, specificandone il tipo, la dimensione e la posizione in memoria. Due funzioni (START e LENGTH) permettono di conoscere anche da programma l'indirizzo di partenza di ogni banco e la sua dimensione in byte. Per il solo banco zero (sprite e bob), LENGTH ritorna il numero di oggetti contenuti.

Il capitolo 6, dedicato alla grafica, ci insegna a usare le istruzioni INK, COLOUR e PALETTE, con le quali è possibile ridefinire i colori di schermo, bordo e tracciatura. GR LOCATE e PLOT posizionano il cursore grafico e disegnano un punto, mentre le funzioni XGR, YGR e POINT danno le coordinate del cursore grafico e il colore alla posizione del cursore. Le istruzioni DRAW, BOX, POLYLINE, CIRCLE e ELLIPSE permettono di disegnare il contorno di semplici figure, mentre BAR POLYGON e PAINT ci danno la possibilità di disegnare figure piene. Le istruzioni SET LINE, SET PATTERN e SET PAINT permettono di definire rispettivamente il tipo di linea di disegno, il pattern per le figure piene e attivare/disattivare il modo OUTLINE per le figure piene. In particolare l'istruzione SET PATTERN si rivela molto potente, infatti, oltre a avere 34 tipi di retino predefinito, è possibile prelevare la matrice da usare per il pattern da un qualsiasi sprite definito dal programmatore. GR WRITING permette di scegliere e combinare il modo di tracciatura tra i quattro modi permessi dall'Amiga: JAM 1, JAM 2, COMPLEMENT e INVERSVID. CLIP infine limita le operazioni grafiche a una regione più piccola dell'intero schermo: le istruzioni grafiche impartite all'esterno dell'area definita con CLIP saranno semplicemente ignorate.

Nel settimo capitolo, dove vengono trattate le strutture di controllo, si inizia a parlare del famigerato GOTO, del quale, forse più per pigrizia che per comodità, nessuno vuol fare a meno. Possiamo osservare come anche *AMOS* sia ben fornito di strutture per il controllo del flusso del programma. Esiste infatti la classica subroutine chiamata con GOSUB, per la quale è prevista l'istruzione POP, che libera lo stack dell'indirizzo di ritorno, permettendo al flusso di programma di uscire da una subroutine per saltare in qualsiasi punto. E a questo punto, i soliti benpensanti staranno già gridando allo scandalo, e purtroppo a ragione, in quanto pur permettendo un'altissima libertà di programmazione, questo tipo d'istruzioni permette anche di generare delle madornali strambate (strutturali si intende), alla faccia di tutto il gran parlare della *buona* programmazione strutturata. Comunque proseguiamo, per niente scandalizzati, e analizzando la sintassi di GOTO e GOSUB notiamo che queste possono riferirsi ai numeri di linea, alle etichette, e per di più al risultato di un'espressione, qualsiasi esso sia. Mi spiego: oltre alle classiche Goto 100 o Gosub LABEL, *AMOS* è capace d'interpretare correttamente un'istruzione tipo Gosub "LABEL"+Str$(2+1)–Chr$(32) che equivarebbe scrivere Gosub LABEL3.

Le classiche IF...THEN...ELSE..., o nella versione strutturata IF...ELSE...ENDIF, permettono la verifica di determinate condizioni, mentre per i cicli possono essere usate le coppie FOR NEXT, WHILE WEND, REPEAT UNTIL, DO LOOP. Da qualsiasi ciclo è possibile uscire tramite le EXIT e EXIT IF, che con con un'opportuna sintassi permettono di uscire anche da più cicli annidati in un colpo solo. Naturalmente esiste la possibilità di eseguire una parte di programma in relazione al valore di una variabile nel formato classico ON GOTO, ON GOSUB e ON PROC, mentre l'esecuzione temporizzata di subroutine e procedure è permessa dalle EVERY n GOSUB ed EVERY n PROC.

Il capitolo 8, dedicato a testi e finestre ci mette a

disposizione un ampio set d'istruzioni per impostare i colori di penna, sfondo e modo di scrittura, che possono essere diversi da quanto previsto per le operazioni grafiche. In questo caso infatti le istruzioni che hanno valore su quanto viene stampato con PRINT sono PEN, PAPER, INVERSE, UNDER, SHADE e WRITING, grazie alle quali è possibile definire il colore della penna, dello sfondo, settare i modi reverse, sottolineato e SHADE che altro non è che la tipica scrittura "fantasma" che viene per esempio utilizzata da Intuition nei titoli delle finestre non selezionate. Una nutrita serie d'istruzioni permette di manipolare il cursore, che può essere spostato in una determinata posizione, oppure possono essere eseguiti spostamenti relativi. Con una serie di funzioni, oltre che a leggere la posizione del cursore, possiamo ottenere delle stringhe composte da caratteri di controllo per lo spostamento dello stesso. Le coppie di funzioni XTEXT, YTEXT e XGRAPHIC, YGRAPHIC permettono di convertire le coordinate grafiche in formato testo e viceversa. Con l'istruzione SET CURS è possibile modificare a piacere l'aspetto del cursore, CURS ON/OFF accende e spegne il cursore, mentre con MEMORIZE è possibile salvare la posizione del cursore (senza usare variabili), per ripristinarla in seguito con REMEMBER. È possibile eseguire lo scroll del testo di una linea o di una sezione di schermo in una determinata direzione, sempre comunque di 8 pixel per volta, tanti quanti sono previsti dalla fonte utilizzabile con PRINT.

La gestione delle finestre è completamente differente da quanto ci si può aspettare dall'Amiga. Innanzitutto è necessario precisare che tutte le operazioni possono essere eseguite sugli schermi, per cui sono veramente pochi i casi in cui si sente la necessità di ricorrere alle finestre. È comunque possibile aprire e chiudere finestre con diversi tipi di bordo, che possono essere spostate e la cui dimensione può essere ridefinita. Sul bordo superiore e inferiore può essere visualizzato un testo ed è possibile scegliere la fonte carattere da utilizzare in ogni window, la cui dimensione dev'essere comunque rigidamente 8 x 8 e non dev'essere proporzionale. Grazie a tre potenti istruzioni (SET SLIDER, HSLIDER e VSLIDER) è possibile dotare ogni programma di comode barre di scorrimento personalizzate, il cui controllo è veramente facile ed efficiente. Alcune istruzioni permettono di utilizzare qualsiasi tipo di fonte di testo con l'istruzione TEXT, che stampa in qualsiasi posizione.

Il nono capitolo riporta la sintassi di qualche decina di funzioni matematiche e trigonometriche, tra le quali notiamo DEGREE e RADIAN che informano *AMOS* dell'unità di misura (gradi o radianti) che si vuole utilizzare per gli angoli.

Il capitolo 10 si occupa degli schermi, per i quali *AMOS* prevede davvero tutto. Possono essere aperti infatti schermi in qualsiasi risoluzione, anche HAM, Extra Half Brite, Dual Playfield e in Double Buffering. Quest'ultima è una tecnica che permette di utilizzare due schermi, di cui uno definito "logico", invisibile, e uno "fisico", quello visualizzato. In questo modo, la tracciatura della grafica viene eseguita sullo schermo logico, e solo a operazioni completate viene effettuato lo SWAP tra i due schermi, ed è questa tecnica che permette animazioni molto fluide, esenti da sfarfallii. Inoltre, memoria permettendo, è possibile aprire schermi ben più grandi di quanto la risoluzione di un monitor permetta, e grazie alle istruzioni SCREEN DISPLAY e SCREEN OFFSET visualizzare la parte interessata.

È possibile avere fino a 8 schermi aperti contemporaneamente, e con una semplice istruzione caricare una schermata grafica. Se lo schermo indicato come parametro non è ancora stato aperto, è *AMOS* stesso che si preoccupa di aprirlo nelle dimensioni corrette per il grafico da caricare, e a load avvenuto, alcune variabili di sistema conterranno le specifiche dello schermo appena aperto. Con la stessa facilità è possibile salvare uno schermo.

Non mancano istruzioni che permettono di spostare uno schermo davanti o dietro agli altri, di farne una copia, di nasconderlo temporaneamente alla vista e rivisualizzarlo. Con l'istruzione SCREEN COPY è possibile copiare il contenuto di tutto o parte di uno schermo su di un altro (o parte di uno schermo in un'altra posizione dello stesso schermo).

L'istruzione DEF SCROLL permette di definire fino a sedici aree con un determinato scroll, che sarà eseguito dal comando SCROLL x, dove x è una delle zone definite. WAIT VBL (Vertical BLank) congela l'esecuzione del programma fino a che il pennello elettronico non completa il ciclo corrente, rendendo possibile una perfetta sincronizzazione video. Una serie d'istruzioni particolari mette a disposizione simpatici effetti come la dissolvenza. APPEAR invece offre la possibilità di scoprire un'immagine gradualmente (immaginatevi una veloce nevicata di pixel che vanno a creare l'immagine), FLASH abilita e disabilita il lampeggio di un determinato colore, mentre SHIFT ruota i colori della palette in un senso o nell'altro. Con SET RAINBOW infine è possibile creare spettacolari sfondi multicolore agli schermi. Non contento ancora, François ha dotato *AMOS* del comando ZOOM, col quale è possibile ingrandire una porzione di schermo. E per chi ancora non si accontentasse degli effetti previsti, *AMOS* mette a disposizione alcuni comandi Basic grazie ai quali è persino possibile programmare il COPPER!

Nei capitoli 11 e 12, prendiamo confidenza con la gestione di sprite hardware, sprite virtuali e bob. Qui apprendiamo che *AMOS*, grazie a particolari tecniche ci dà la facoltà di gestire sprite ai quali possiamo assegnare più di 16 pixel di larghezza (fino a un massimo di 128) e più di quattro colori. Questa tecnica, che dà la possibilità di combinare più sprite tra loro per ottenere immagini col numero di colori e della larghezza voluta, è estremamente impegnativa, tanto che fino a ora è stata utilizzata esclusivamente in linguaggio Assembly. Ebbene, *AMOS* gestisce tutto il processo automaticamente e in modo assolutamente trasparente al programmatore.

Delle 29 istruzioni a disposizione ci colpisce la comoda GET SPRITE PALETTE, che copia la palette utilizzata dall'oggetto (sprite o bob) nella palette dello schermo corrente. Inoltre, GET SPRITE e GET bob permettono di convertire una parte di schermo nell'oggetto corrispondente. PUT bob, PASTE bob invece permettono di copiare il disegno dell'oggetto sullo schermo. LIMIT bob invece restringe l'area visibile per i bob.

Il capitolo successivo si occupa del controllo degli oggetti e del mouse, che è ridefinibile a piacere con una forma predisposta, oppure con un qualsiasi sprite utente. Con HIDE e SHOW si può facilmente "spegnere e riaccendere" il puntatore. LIMIT MOUSE permette di limitare il movimento del mouse a un'area ristretta, e una serie di funzioni ci dà la possibilità di controllare

l'azione sui tasti del mouse e le coordinate del puntatore, che possono anche essere impostate oltre che lette. Ciò significa che con poche istruzioni, possiamo comandare il puntatore del mouse leggendo per esempio gli spostamenti di un joystick per il quale sono riservate ben sei funzioni.

Alcune istruzioni permettono la gestione delle collisioni tra sprite, tra bob, oppure tra sprite e bob. Altre istruzioni danno la possibilità di definire porzioni di schermo rettangolari, più o meno grandi, grazie alle quali è possibile gestire le collisioni tra le stesse e gli oggetti, mouse compreso.

Il quattordicesimo capitolo con ben 30 pagine ci porta a contatto della caratteristica forse più appariscente di *AMOS*: l'animazione. Se con tutti gli altri linguaggi, la costruzione anche di una breve animazione non è opera di poco conto, costruire animazioni con *AMOS* risulta di una facilità estrema. Questo capitolo descrive nei dettagli AMAL (AMOS Animation Language), un vero e proprio linguaggio per la gestione e il controllo delle animazioni "dentro" l'interprete stesso. AMAL, con le sue 13 potenti istruzioni e un numero poco più alto di funzioni, permette di muovere e animare gli oggetti (assegnare a determinati intervalli una determinata immagine a un oggetto, bob o sprite), eseguire cicli, test, salti incondizionati, controllare il joystick, il mouse, usare particolari registri interni come fossero variabili... I programmi AMAL, vengono eseguiti durante gli interrupt, rubando pochi cicli operativi al programma *AMOS*, che continua indisturbato il suo lavoro.

Possono essere eseguiti fino a sedici programmi di animazioni contemporaneamente, e con una tecnica opportuna, che scavalca gli interrupt di sistema dell'Amiga, il numero di animazioni contemporaneo sullo schermo può salire fino a 64.

Le istruzioni del programma di animazioni di regola devono essere assegnate a una stringa, che, con con l'istruzione "AMAL x,A$" viene associata al programma "x" e compilata (trasformata in puro LM) all'esecuzione del programma *AMOS*. Alcune istruzioni Basic permettono durante l'esecuzione del programma di far partire uno o più programmi di animazione, fermarli, congelarli... È persino possibile associare le animazioni agli schermi, per ottenere con una manciata d'istruzioni spettacolari e velocissimi scroll hardware, che a molti faranno rimpiangere il tempo speso ad apprendere il C o l'Assembly.

Il capitolo successivo spiega il modo per ottenere dozzine di sfondi grafici per i nostri programmi con relativamente pochi elementi di base e ci mette a disposizione alcune potenti istruzioni per la manipolazione di porzioni di schermo. Per quanto riguarda la gestione dei BLOCK, sottolineamo la possibilità di essere salvati in memoria anche in formato compattato.

Il capitolo 16 è nuovamente fonte di meraviglia, per la potenza e facilità che ci riserva *AMOS* nella creazione dei menu. Chi si era abituato alle proverbiali ristrettezze dell'Amiga BASIC andrà senza dubbio in visibilio: *AMOS* infatti anche per i menu permette più o meno *tutto* quello che potenzialmente è possibile ottenere con il C. I menu possono essere ramificati gerarchicamente fino a otto livelli e a ogni item può essere associata una qualsiasi combinazione di tasti. I menu possono essere dotati della classica barra o meno, possono essere posizionati in qualsiasi punto sullo schermo, può essere previsto che i vari item siano collocabili dall'utente finale nella posizione che più gradisce...

È possibile modificare la fonte e lo stile dei caratteri, inserire grafica colorata e infine è possibile fare in modo che durante l'attivazione di un oggetto venga automaticamente e continuamente chiamata una procedura. Questo permette la modifica dinamica dei menu, e la visualizzazione di qualsiasi animazione nell'item attivo!

L'utility MENU EDITOR permette di costruire con relativa facilità complessi menu che possono essere salvati per essere in seguito utilizzati dal programma che se ne servirà, caricandoli in un apposito banco di memoria. In tal modo nel programma finale si risparmia tutta la parte dedicata alla creazione dei menu.

Il diciassettesimo capitolo ci insegna come utilizzare le istruzioni che similmente all'Amiga BASIC permettono di produrre suoni e musichette più o meno simpatiche. In più, *AMOS* mette a disposizione numerosi comandi che danno la possibilità di eseguire in interrupt le numerosissime composizioni musicali presenti sul mercato PD e composte con *SoundTracker*, *Sonix* e *GMC*. Queste devono essere preventivamente trasformate in un formato interpretabile da *AMOS* con alcune utility che vengono fornite nei tre dischetti. Inoltre, è possibile utilizzare la vasta biblioteca strumenti campionati che possono anche essere associati a una voce dell'Amiga, per essere utilizzati con la tradizionale PLAY. Da notare che il controllo del volume è separato: MVOLUME regola il volume dei moduli musicali, e VOLUME regola il volume delle istruzioni tradizionali. Ciò rende facile la realizzazione di programmi con una musica di sottofondo a basso volume, per dare più rilevanza agli

La velocità di AMOS

Purtroppo il programma scelto per testare la velocità di *AMOS* non è tra i più felici infatti si può notare che a causa della particolare gestione che fa *AMOS* dei parametri passati col comando BAR (che disegna un rettangolo pieno), è necessario integrare il programmino con tre test IF-THEN, che rallentano notevolmente l'esecuzione. Ebbene, nonostante ciò i tempi che si ottengono sono persino inferiori al tempo impiegato dal velocissimo *GFA* compilato (si veda *Commodore Gazette* 4/90)

[illegible]	Secondi
[illegible]	20,57
[illegible]	[illegible],46
[illegible]	7,16
[illegible]	6,10
[illegible]	5,04

[illegible]

Ultimissime su AMOS

Al momento di andare in stampa riceviamo dalla AP&S il disco "AMOS 1.2 Update" con il quale è possibile aggiornare *AMOS* alla versione 1.2. Abbiamo quindi deciso d'inserire questo riquadro per informare i lettori delle appetitose novità.

Il disco in questione è di pubblico dominio, e ne è incoraggiata la diffusione, va però tenuto presente che il copyright rimane inalterato sia sulla versione 1.1 sia sull'upgrade 1.2.

Il disco contiene il file eseguibile AMOS_Update_V1.2 che modifica una copia del disco programma di AMOS, la quale dev'essere preventivamente predisposta. Il disco risultante conterrà il nuovo modulo di run-time RAMOS1.2, il nuovo interprete AMOS1.2 e la nuova versione dei programmi CONFIG1_2.AMOS e MENU_EDITOR.AMOS. Oltre a ciò troveremo un ulteriore programma completamente nuovo, LOCK.AMOS, che se opportunamente eseguito permette di chiudere permanentemente le procedure di un nostro programma AMOS (non potranno in alcun modo essere riaperte, un po' quello che succede con l'opzione "P" del save dell'Amiga BASIC, che non permette di listare il programma) proteggendo il codice in esse contenuto da occhi indiscreti. A questo riguardo il simpatico Lionet lancia un allettante sfida: promette una bottiglia di ottimo champagne francese al primo programmatore che riuscirà a scrivere un programmino (naturalmente in AMOS) capace di eliminare l'insolita protezione creata con LOCK.AMOS.

La lettura del lungo READ_ME (quasi 25000 caratteri) ci informa del massiccio lavoro di debug eseguito, nonché delle molteplici migliorie e aggiunte apportate a tutto il sistema. A causa di ciò le dimensioni del già poderoso interprete passano da 87 a circa 133K! L'incremento delle dimensioni del modulo RAMOS è invece (per fortuna) più limitato.

Una modifica sostanziale riguarda il selettore dei file, che pur essendo molto curato nella versione 1.1 diventa ora "ARP like". Infatti, un item può essere selezionato anche se la lettura del dispositivo corrente non è ancora terminata, rendendo più veloci le operazioni verso le memorie di massa. I gadget dei dispositivi presenti nel sistema che nella versione 1.1 venivano mostrati sul lato sinistro sono scomparsi nella versione 1.2 e la lista completa dei dispositivi esistenti, (anche di quelli creati con ASSIGN), viene mostrata con la pressione del tasto destro del mouse.

Due nuove funzioni, DEV FIRST$() e DEV NEXT$(), permettono ora di conoscere rispettivamente il primo dispositivo della lista e il successivo a quello corrente.

Grazie ad altre tre nuove funzioni, HREV(), VREV() e REV(), oppure a una particolare sintassi da usarsi nei comandi che fanno riferimento a sprite e bob, AMOS 1.2 è in grado di eseguire il "flip" automatico in tutte le direzioni (orizzontale, verticale, o entrambi contemporaneamente) sia degli sprite che dei bob. Questa possibilità fa risparmiare una notevole quantità di memoria, infatti nelle sequenze d'immagini da usare in ogni animazione quasi sempre è necessario ricorrere alla stessa immagine ma ruotata o invertita. Con *AMOS 1.2* non è necessario avere in memoria tutta la sequenza, comprendente anche le immagini invertite o ruotate, le quali possono essere generate automaticamente con estrema facilità. La stessa tecnica può essere applicata a sezioni di schermo (BLOCK), che possono essere manipolati con HREV BLOCK e VREV BLOCK.

Un'ulteriore caratteristica di cui onestamente si sentiva la necessità, e che ora è presente, è lo swap dei banchi di memoria. Con BANK SWAP il programmatore ha ora la possibilità di avere a disposizione più di un banco di memoria per ogni genere di dati: sprite, musica, icone. Lo stesso file READ_ME ci informa tra l'altro di una semplice modifica da apportare al programma di conversione dei moduli da SoundTracker ad AMOS, grazie alla quale viene eliminato un fastidioso inconveniente che rendeva i moduli così generati non perfettamente eseguibili.

L'impareggiabile François conclude con alcune anticipazioni dalle quali apprendiamo che oltre al compilatore d'imminente uscita, (probabilmente nel momento in cui questo numero di *Commodore Gazette* sarà in edicola *AMOS Compiler* sarà già disponibile) verrà presto pubblicato un ulteriore upgrade che doterà *AMOS* di un'interfaccia AREXX compatibile. Inoltre in questa prossima release l'estensione COMMS renderà il linguaggio capace di gestire direttamente sia la porta parallela che la porta seriale RS232.

Con la versione 2.0 di *Kickstart* e *Workbench*, mentre *AMOS 1.1* pur funzionando perfettamente si bloccava nel momento in cui veniva chiamato il selettore di file, *AMOS 1.2* non sembra denunciare alcun malfunzionamento, rivelando anzi un incremento di fluidità nelle animazioni. Infatti, un'animazione che con la versione 1.1 dava qualche leggerissimo sfarfallio di tanto in tanto (anche sulla nuova versione del sistema operativo) con *AMOS 1.2* risulta invece perfettamente stabile.

Concludiamo sottolineando che *AMOS 1.2* contempla in pieno il multitasking, a differenza della versione precedente in cui ne [illegible] limitato.

effetti sonori legati all'azione in corso. Naturalmente, non manca l'istruzione LED che permette di controllare il filtro e le istruzioni SAY e SET TALK.

Nel capitolo 18 riservato alla gestione dell'input da tastiera notiamo la presenza d'istruzioni e funzioni che ci permettono d'interpretare qualsiasi tasto o combinazione di tasti premuta, mentre con KEY SPEED possiamo regolare la velocità di ripetizione dei tasti.

Il manuale prosegue con l'esame di una manciata di altri comandi, alla rinfusa (PRINT, REM, DATA...) per arrivare al capitolo 20 dove sono esaminate le numerose istruzioni legate alle memorie di massa. Queste permettono di eseguire più o meno tutte le principali operazioni sui file, compresa la creazione di una directory, il rename, la selezione di un percorso... La gestione dei file prevede anche file con accesso a indice o casuale. Pur essendo prevista l'apertura di un canale di comunicazione con la porta seriale RS232, non sono però previste opzioni per i parametri.

Nel penultimo capitolo vengono illustrate alcune istruzioni sulla compattazione e scompattazione degli schermi, grazie alle quali è possibile risparmiare una notevole quantità di memoria sui dischi. Gli algoritmi utilizzati supportano tutti i tipi di schermi (anche HAM).

Siamo finalmente giunti all'ultimo capitolo, dove veniamo messi al corrente delle tecniche d'interfacciamento di *AMOS* con l'Assembly. Oltre alle istruzioni per leggere e scrivere direttamente in memoria, PEEK e POKE per i byte, e le opportune varianti per le word e le long word, troviamo un'istruzione che permette di copiare zone di memoria, un'altra ci permette di riempire un'area di memoria con un determinato pattern, mentre la funzione HUNT verifica l'esistenza di una determinata stringa in memoria. Con alcune istruzioni possiamo manipolare i byte e anche singoli bit.

PLOAD carica in un banco di memoria un file binario e CALL cede il controllo a una routine in LM permettendo anche il passaggio di parametri, mentre i comandi/funzioni AREG e DREG danno la possibilità di leggere o scrivere direttamente nei registri del 68000.

Tutte le funzioni di libreria residenti in ROM (DOS, EXEC, GRAPHICS e INTUITION) sono direttamente accessibili da *AMOS*, mentre l'accesso alle librerie esterne è possibile con le tecniche usuali.

Conclusioni

Finalmente esiste un linguaggio di altisimo livello, con il quale si riescono a ottenere notevolissimi risultati. Forse non è ancora del tutto esente da piccoli bug, è ancora perfezionabile e completabile, e non funziona perfettamente sull'Amiga 3000, ma tenendo conto della sua giovanissima età e delle applicazioni per cui viene consigliato, si rivela sicuramente un prodotto di un livello nettamente superiore a tutti gli altri Basic disponibili. È chiaro che se ci stiamo accingendo a scrivere un programma gestionale, è forse meglio rivolgere l'attenzione verso altri orizzonti, ma per quanto riguarda grafica, animazione e suono non esiste al momento migliore alternativa.

Per ulteriori informazioni contattare direttamente:

Applied Peripherals & Software
(AMOS della Mandarine Software, L. 130.000 Iva compresa)
Via Papa Giovanni, 37 - 33040 Corno di Rosazzo (UD)
(Tel. 0432/759264 - Fax 0432/759264)

PROVE SOFTWARE

KICKSTART E WORKBENCH 2.0: UN PASSO AVANTI

Un look che sta diventando sempre più professionale per un Amiga a sua volta sempre più "grande". L'evoluzione continua...

di Alfredo Prochet

Ormai tutti in trepidante attesa della versione 1.4 del sistema (e già correvano voci a riguardo della 1.5), siamo rimasti abbastanza sorpresi nel venire a sapere che la nuova "release" del *Kickstart* sarebbe stata etichettata invece 2.0. Non appena, però, l'abbiamo potuta esaminare a fondo, in versione pressoché definitiva, sul nuovo Amiga 3000, abbiamo dovuto convenire che questa scelta era, in effetti, assolutamente obbligatoria. Tali e tante sono le novità apportate nel sistema 2.0 che non basterebbe l'intera rivista a menzionarle tutte. Basti pensare al fatto che le dimensioni stesse del *Kickstart* sono raddoppiate: 512K contro i 256K delle versioni precedenti.

Sebbene a un primo sguardo i cambiamenti sembrino soprattutto di carattere estetico o comunque soprattutto a livello di *Workbench*, le vere novità sono molto più "in basso", e coinvolgono librerie, dispositivi, risorse e finalmente, ed era ora, lo stesso AmigaDOS.

Procediamo però con ordine, e iniziamo gradualmente un primo esame dei cambiamenti più evidenti (in successivi articoli vedremo di approfondire gli argomenti più interessanti e complessi). Iniziamo pertanto dal *Workbench*, che, come sempre, costituisce il "primo contatto" con la macchina.

Il Workbench

Lo schermo del *Workbench* è ora interlace per default (anche se, grazie alle meraviglie hardware del 3000, è facile non accorgersene), ma è possibile, e consigliabile, per chi non possedesse un monitor multisync come l'ottimo 1950, riportarlo a una normale risoluzione 640 x 256; sebbene, come vedremo, le scelte al riguardo siano molteplici. Partendo dall'alto troviamo già un cambiamento nella barra del titolo: in grigio su nero il *Workbench* ci comunica infatti quanta memoria è disponibile, suddividendola però in "graphics mem" (la consueta memoria chip) e "other mem" che rappresenta invece la memoria fast. Ciò consente di rendersi conto con maggiore efficacia e immediatezza di come vengono utilizzate le risorse del sistema dai programmi.

Appena sotto la barra del titolo scopriamo che il nostro *Workbench* è diventato in tutto e per tutto una normale finestra, che può quindi essere ridimensionata e spostata davanti e dietro le altre in maniera da avere sempre a portata di mano le icone dei drive, che, nelle versioni precedenti, a volte venivano irrimediabilmente oscurate dalle finestre di altri programmi. È comunque prevista la possibilità di tornare a un "normale" *Workbench* backdrop, per chi lo desiderasse. La novità più evidente del nuovo sistema, in questa come in tutte le altre finestre, è però costituita dal loro aspetto. Il "look" dell'intero sistema è stato quasi completamente ridisegnato e i risultati si vedono, eccome. Innanzitutto i violenti colori di gusto molto "americano" usati fino a oggi per default sono stati finalmente abbandonati. I colori principali ora sono un riposante grigio di sfondo, il bianco, il nero, e un sobrio azzurro carta che mette in risalto, senza offendere l'occhio, tutto ciò che va messo in risalto. Grazie a un ottimo uso di questi colori e a un pizzico di restyling, le finestre e le icone al loro interno hanno assunto un piacevole e ottimamente calibrato aspetto tridimensionale, rendendo così gli schermi e le finestre di Intuition veramente "intuitivi".

Cambiamenti anche nei gadget di sistema: la coppia window to front/window to back viene sostituita da un unico gadget che svolge alternativamente le due funzioni: selezionato una volta, porta la finestra in primo piano, selezionato una seconda volta, la rimanda sullo sfondo. Ci vuole un attimo per abituarsi, ma poi ci si chiede con stupore come mai per tutti questi anni di gadget ne sono occorsi due. Al posto del vecchio "window to back", invece, è ora presente il nuovo gadget zoom. Esso fa variare le dimensioni ed eventualmente anche la posizione della finestra tra due serie di valori prestabiliti, dipendenti dalla finestra stessa. Per esempio, le finestre del *Workbench* vengono alternativamente ridotte a dimensioni minime e quindi riportate alla

grandezza normale tramite la selezione di questo gadget, mentre la finestra dello shell viene portata invece a occupare l'intero schermo. Un particolare gradevole riguardo a questo nuovo gadget è che appare anche nei programmi precedenti il *Kickstart 2.0*. La sua presenza è infatti legata a quella del gadget di ridimensionamento: se il flag WINDOWSIZING risulta impostato all'apertura di una nuova finestra, entrambi i gadget saranno infatti presenti.

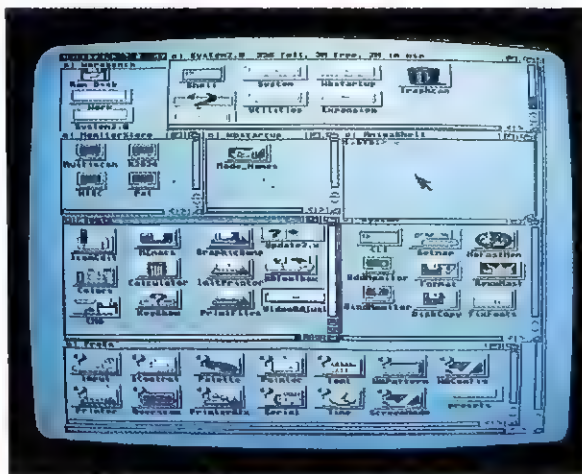
Poche novità, invece, per quanto riguarda i gadget tipici delle finestre del *Workbench*. Le barre di scorrimento e le freccine relative hanno solamente cambiato un po' di aspetto e di posizione, ma il funzionamento è sempre lo stesso. È sparito invece l'indicatore analogico dello stato di riempimento di un disco (la cosiddetta "barra del carburante") sostituita da più complete ed esaurienti informazioni poste di seguito al titolo della finestra: il numero di file e directory contenuti (anche se non visibili come icone) e il numero di blocchi utilizzati sul disco.

Alcune novità sono presenti anche all'interno delle finestre stesse. È possibile infatti selezionare contemporaneamente più icone semplicemente tracciando intorno a esse un rettangolo con il mouse (un po' come avviene nell'inventario delle avventure della ICOM, per intenderci). Inoltre, in un gruppo di icone selezionate, è possibile aggiungere o rimuovere singole icone in maniera molto semplice.

I menu del Workbench

Per scoprire però dove sono i veri cambiamenti del *Workbench*, diamo una rapida occhiata ai suoi menu, più che triplicati come dimensioni. Essi sono disposti in ordine gerarchico. Al livello più alto troviamo il menu le cui funzioni influiscono sull'intero *Workbench*; il suo titolo è, abbastanza prevedibilmente, "Workbench" e al suo interno troviamo innanzitutto la voce Backdrop, che consente di riportare la finestra del *Workbench* sullo sfondo, inamovibile com'è stata fino a oggi. Execute Command permette invece di eseguire qualunque comando senza bisogno di attivare uno shell, fornendo anche una finestra in cui visualizzare un eventuale output. Redraw All, Last Error e Version non sono una novità, ma la sono invece Update All e Quit. Il primo fa sì che il *Workbench* aggiorni immediatamente il contenuto di tutte le sue finestre in maniera da riflettere eventuali cambiamenti avvenuti sui dischi ma al di fuori del suo controllo, per esempio file e directory cancellati o aggiunti tramite uno shell. Quit, invece, o più semplicemente il gadget di chiusura della finestra principale del *Workbench*, consentono, se non è presente alcuna applicazione lanciata dal *Workbench* stesso, di "chiuderlo" completamente, rimuovendolo dal sistema e liberando così tutta la memoria da esso occupata. Bisogna però fare attenzione e conservare almeno uno shell o un altro mezzo per poter interagire con il sistema, per non essere "tagliati fuori" del tutto e costretti a ricorrere ai tre tasti magici del reset.

Il secondo menu, Window, è dedicato invece alle singole finestre. New Drawer rappresenta in pratica l'equivalente del comando CLI Makedir, ed evita finalmente lo scomodo e poco elegante passaggio di dover ricorrere alla copia e alla ridenominazione del famoso drawer Empty. Un'altra voce nuova è Open Parent, che porta in primo piano, se necessario riaprendola, la finestra genitrice di quella correntemente attiva. Update rappresenta una versione ridotta di Update All, limitata alla sola finestra selezionata. Select Contents consente di selezionare contemporaneamente tutte le icone di una finestra, mentre Clean Up, nel riordinare il contenuto di una finestra, tiene anche conto delle dimensioni del nome delle varie icone, evitando antiestetiche sovrapposizioni. L'operazione di Snapshot, ora, permette di scegliere se agire sulle sole dimensioni e posizione di una finestra o anche su tutte le icone in essa contenute. Le novità più grosse di questo menu, però, vengono dalle ultime due voci: Show e View By. La prima consente di visualizzare nelle finestre del *Workbench* anche i file e le directory sprovvisti di icone, fornendogliene di temporanee. La seconda, invece, consente di determinare come dev'essere visualizzato il contenuto della finestra: se normalmente tramite immagini, oppure tramite il solo nome dei file e delle directory, mediante un elenco molto simile all'output del comando List (ma sul quale è possibile agire direttamente con il mouse, come se si trattasse di icone), elenco eventualmente ordinato per nome, data o dimensioni. Visto che le caratteristiche di questo genere assegnate a una finestra vengono ereditate automaticamente da tutte le sue eventuali discendenti, diventa così veramente molto comodo esplorare il contenuto di qualunque disco senza dover ricorrere al CLI.

Il terzo menu, Icons, è invece dedicato alla manipolazione diretta delle singole icone e dei dati e programmi a esse associati. Open, Copy e Rename non sono mutati, ma è stato invece mutato Information, che sostituisce la voce Info delle versioni precedenti. Oltre a non dover più dipendere dalla presenza della info.library, il cui equivalente si trova ora direttamente nel *Kickstart*, è stato completamente ridisegnato, anche se, in effetti, l'unica caratteristica veramente nuova è quella di poter accedere direttamente ai bit di protezione dei file. In questo menu troviamo nuovamente un'opzione di Snapshot, accompagnata però dalla nuova UnSnapshot il cui compito è proprio quello di "liberare" la posizione di un'icona, lasciando il compito di trovarle una collocazione appropriata al buon gusto del *Workbench* stesso. Anche qui, però, le vere novità sono al fondo. Si tratta di Leave Out e Put Away, tra loro complementari. Queste due funzioni permettono d'inserire in maniera permanente nella finestra del *Workbench* icone provenienti da altri dischi o directory in modo da averle sempre a disposizione senza bisogno di "scavare" nei meandri dell'hard disk alla loro ricerca. Ancora nel menu Icon troviamo

Delete, che sostituisce la vecchia opzione Discard; Format Disk, e infine Empty Trash, tutti comandi già conosciuti.

L'ultimo menu, Tools, è invece un po' speciale: a esso infatti i programmi applicativi scritti da terzi possono agganciare, tramite apposite procedure, nuove voci e nuove funzioni, al fine di migliorare e completare ulteriormente lo spesso sottovalutato ambiente *Workbench*. Al momento, questo menu contiene solamente la voce ResetWB, che, come si può facilmente intuire, reinizializza l'intero *Workbench*: finestre, colore e risoluzione in maniera da riflettere i parametri correntemente impostati.

Il disco del Workbench

Una volta presa dimestichezza con tutte le novità del *Workbench*, iniziamo a esplorare il contenuto del disco sistema. Nella directory principale, oltre all'icona per l'attivazione dello shell troviamo ben sei sottodirectory: Trashcan ed Expansion non sono mutate se non nell'estetica. Qualche novità la troviamo invece all'interno della directory System. In aggiunta a un ormai inutile CLI (inutile perché è ormai in tutto e per tutto identico allo shell) e ai consueti Format, DiskCopy, Setmap, NoFastMem e FixFonts, quasi tutti senza significativi cambiamenti, troviamo RexxMast, AddMonitor e BindMonitor. Il primo costituisce la parte visibile da *Workbench* del linguaggio ARexx, ora finalmente parte integrante del sistema. AddMonitor e BindMonitor, invece, forniscono il necessario supporto per la gestione di monitor diversi dall'onnipresente 1084. Il primo informa il sistema del tipo di monitor collegato (è necessario inserire in precedenza il file di configurazione relativo a questo monitor nella directory WBStartup), in modo che i programmi interessati possano essere poi informati dal sistema stesso sulle risoluzioni grafiche disponibili. Il secondo programma, BindMonitor, invece, associa al codice numerico interno corrispondente ai vari tipi di monitor una stringa di caratteri, in pratica il nome stesso del monitor, in modo da facilitare l'utente nella scelta.

Nella directory Utilities troviamo invece, insieme a programmi conosciuti come Say, More e Clock, un visore di file ILBM: Display e i programmi del Commodities Exchange, un'altra delle grandi novità di questa release. Si tratta di programmi che, come il conosciutissimo PopCLI, esaminano gli eventi provenienti dai dispositivi esterni (principalmente mouse e tastiera) prima che Intuition li possa "diffondere" al resto del sistema. In tal modo possono riconoscere la pressione di determinati tasti e agire di conseguenza qualunque sia al momento la finestra attiva. Fino alla versione 1.3, però, questi programmi esistevano allo stato "selvaggio", ovvero senza alcun metodo di controllo che impedisse eventuali interferenze dell'uno con l'altro. Con la versione 2.0, tuttavia, è stato introdotto il sistema del Commodities Exchange, nel quale un programma principale, denominato appunto Exchange consente il controllo, la disattivazione o la momentanea sospensione di uno o più di questi programmi, che devono però interfacciarsi con il sistema secondo regole ben precise. Oltre a quello di gestione sono presenti sul disco sistema quattro altri programmi: IHelp che consente di assegnare, a particolari combinazioni di tasti, delle funzioni fino a oggi di dominio esclusivo del mouse, come il restringimento o l'allargamento delle finestre. AutoPoint serve invece a far sì che il semplice spostamento del mouse all'interno di una finestra basti a farla diventare quella attiva senza bisogno di agire sul pulsante sinistro, come avviene per esempio sulle workstation della Sun. Blanker causa il completo annerimento del video trascorso un determinato periodo di tempo senza la pressione di alcun tasto (un'altra delle funzioni cui ci aveva abituato PopCLI). FKey consente di associare stringhe di caratteri ai tasti funzione e NoCapsLock, infine, permette di disabilitare temporaneamente il tasto Caps Lock.

Al posto della ormai inutile directory Empty è presente ora la directory WBStartup. La sua particolarità consiste nel fatto che i programmi al suo interno vengono automaticamente eseguiti dal sistema in fase di apertura del *Workbench*, costituendo così una sorta di prolungamento dinamico della startup-sequence, facilmente modificabile dall'utente. Al suo interno si trova inizialmente il solo file Mode_Names, che contiene i nomi dei monitor da associare al sistema tramite il citato comando BindMonitor, il che avviene, così, in maniera del tutto trasparente all'utente.

Le nuove Preferences

Ma ora, ed è il caso di dire *dulcis in fundo*, veniamo alla directory Prefs, al cui interno, al posto del familiare Preferences, troviamo una vera moltitudine di programmi. Preferences è stato infatti spezzettato in più parti, ognuna delle quali, arricchita e migliorata, è ora un programma a sé stante. Input, per esempio, consente d'intervenire in maniera veramente completa sui parametri di Intuition che governano il comportamento del mouse e della tastiera: velocità ed eventuale accelerazione del mouse, temporizzazione del doppio click, velocità e ritardo nella ripetizione dei tasti possono essere controllati in maniera rapida e immediatamente verificabile. Gli altri programmi comprendono Time, per

I comandi della versione 2.0

L'asterisco indica i comandi nuovi o con modifiche di una certa rilevanza.
La R indica i comandi residenti direttamente nel Kickstart.

		Ed	*	Mount	
		Edit		NewShell	
		Else	R	Path	R
		EndShell	R	Prompt	R
		EndIf	R	Protect	
		EndSkip	R	Quit	R
		Eval		Relabel	
AddBuffers		Execute		RemRAD	
Alias	* R	FailAt	R	Rename	
Ask	R	Fault	R	Resident	* R
Assign	*	FileNote		Run	R
Avail		GetEnv	R	Search	
BindDrivers		IconX		SetClock	
Break		If	R	SetDate	
Cd	R	Info		SetEnv	R
ChangeTaskPri		Install		Skip	R
Copy		IPrefs	*	Sort	
CPU	*	Join		Stack	R
Date		Lab	R	Status	
Delete		List		Type	
Dir		LoadWB		Version	*
DiskChange		Lock		Wait	
DiskDoctor	*	Makedir		Which	
Echo	* R	MakeLink	*	Why	R

l'impostazione della data e dell'ora; Palette, per la modifica dei colori dello schermo; Pointer, per la personalizzazione del puntatore; Printer e PrinterGFX, per il controllo dei parametri di stampa e Serial, per impostare i valori relativi all'interfaccia seriale.

Inoltre vi sono moltissime vere novità, con le quali, almeno per i primi tempi, è realmente difficile smettere di giocherellare. Si tratta, per esempio, di WBPattern, che consente d'inserire dei pattern a piacere al posto del consueto colore di fondo nel *Workbench* e nelle sue finestre. Se si evitano di usare gli orribili pattern predisegnati, si possono raggiungere dei risultati veramente piacevoli. Font permette invece di controllare quale fonte di caratteri utilizzare in tre diverse circostanze: per il testo delle icone, il titolo delle finestre, e nel resto del sistema (shell compresi). Utilizzando, per esempio, la fonte Helvetica o Courier in alta risoluzione interlace, si ottengono risultati esteticamente molto appaganti uniti a un certo look professionale.

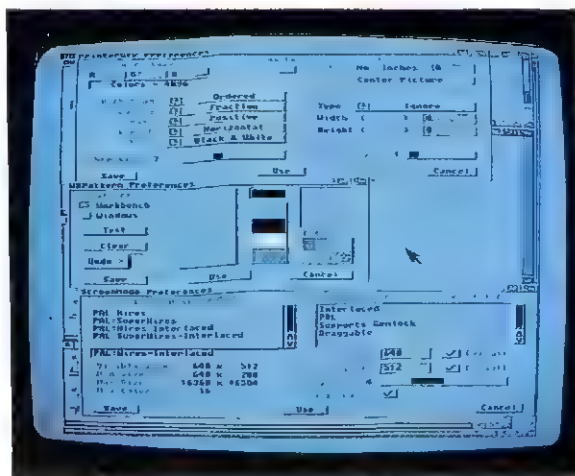
Finalmente è possibile controllare e utilizzare anche l'area ai bordi del video tramite Overscan, grazie al quale si possono definire due diverse aree di overscan: una per la grafica, e una, ovviamente più ristretta, per il testo. Quest'area di overscan viene correttamente utilizzata anche modificando la risoluzione del *Workbench* tramite ScreenMode. Quest'ultimo programma consente infatti di scegliere per lo schermo del *Workbench* la risoluzione desiderata fra tutte quelle disponibili per il tipo di monitor scelto. Con un Amiga 3000 e un monitor multiscan si possono ottenere risultati quali risoluzioni di 1280 x 512 in Superhires o 640 x 960 in Productivity Mode, tutto questo senza contare l'eventuale overscan! È possibile richiedere l'apertura di uno screen più grande dell'immagine visibile, e sarà Intuition stesso a provvedere automaticamente allo scroll quando il puntatore viene portato e "spinto" oltre al bordo dello schermo. Gli ultimi programmi da menzionare nella directory Prefs sono IControl, che consente di stabilire quali combinazioni di tasti devono essere associate da Intuition a operazioni quali il movimento dello schermo del *Workbench* e la cancellazione dei requester di sistema, e infine WBPrefs che consente di decidere se, alla sua apertura, il *Workbench* dev'essere o meno una finestra come le altre (per default lo è) e se attivare un'altra nuova caratteristica della versione 2.0: la possibilità di portare una finestra in primo piano semplicemente tramite un doppio click del mouse al suo interno. Ciò si rivela estremamente comodo quando il gadget di profondità non è visibile o quando non è presente del tutto.

Il disco degli Extras

Sul disco degli Extras, oltre all'immancabile Trashcan, troviamo altre due directory: Tools e MonitorStore. In quest'ultima sono contenuti i file dati relativi ai vari monitor che possono essere collegati all'Amiga. Sono correntemente quattro: PAL, NTSC, A2024 e Multiscan. Nella directory Tools troviamo invece alcuni dei programmi assenti dal disco principale insieme ad altri che sono sempre "vissuti" quaggiù: Calculator, Cmd, Colors, IconEdit, MEmacs (sempre più "amighevole"), KeyShow (il vecchio KèyToy), InitPrinter, PrintFiles e GraphicDump.

Sempre sul disco degli Extras notiamo, per la sua assenza, l'Amiga BASIC; e infine, non visibili da *Workbench* (a meno di non selezionare View All Files), le directory fonts e devs. La prima contenente le consuete fonti, la seconda i driver delle varie stampanti e le keymap internazionali. Specialmente in questa versione dove è possibile fare un uso veramente estensivo delle fonti, l'assegnazione della directory logica FONTS: a questo disco diventa, se non si possiede un hard disk, assolutamente d'obbligo.

Lo shell e comandi dell'AmigaDOS

Veniamo ora però alla parte che forse più interesserà gli utenti più esperti: CLI e comandi dell'AmigaDOS. Il CLI vero e proprio non esiste più: grazie all'inclusione in ROM dei vari Shell-Seg e Newcon-Handler, ora tutte le finestre aperte tramite le icone Shell e CLI, o con i rispettivi comandi NewShell e NewCLI (mantenuto per compatibilità con le versioni precedenti), sono degli shell, con tutti i vantaggi che da ciò derivano. Oltre alle facilitazioni che già conosciamo: la comodità di editing, la "command history" e l'uso degli "alias", nella nuova versione si sono aggiunte altre caratteristiche che rendono ancora più interessante e rapido il lavoro tramite shell. La prima che andiamo a esaminare viene denominata "character-mapping": a ogni shell viene associato un buffer in cui vengono automaticamente memorizzati i caratteri in essa visualizzati. In tal modo, mutando le dimensioni della finestra, non ne viene più perso il contenuto come accadeva in precedenza, ma il testo al suo interno viene dinamicamente riadattato ai nuovi confini. Ciò si rivela fondamentale per il funzionamento della seconda grande novità: il Cut & Paste tra finestre di tipo console. In maniera analoga a quanto accade nella maggior parte dei word processor, tramite la semplice selezione con il mouse a pulsante sinistro premuto è possibile delimitare un'area di testo e copiarla quindi mediante la pressione dei tasti Amiga destro-V in un'altra finestra di tipo console. Ciò rende possibile, per esempio, portare rapidamente l'output del comando list all'interno di un editor come *TxEd+*, elaborarlo come desiderato, salvare il file così ottenuto, e quindi eseguirlo come script. Bisogna dire al riguardo, però, che il programma di pubblico dominio *Snap*, che svolge una funzione analoga, si dimostra ancora superiore. Dal momento che riconosce i caratteri direttamente dal video tramite la loro forma e non li preleva da un buffer, può infatti leggere blocchi di testo provenienti

da ogni tipo di finestra e quindi operare direttamente il passaggio da editor a shell, cosa che con il *Kickstart 2.0* non è invece possibile. Altra piccola novità dello shell è il gadget di chiusura, che si comporta in maniera analoga al comando EndShell. Un terzo modo per chiudere uno shell è la pressione dei tasti Ctrl-\ (il carattere di fine file).

Ed eccoci finalmente all'esame della directory c e dei comandi in essa contenuti. Sorpresa! Sembra mezza vuota. Ma è solo apparenza: infatti gran parte dei comandi più usati e meno voluminosi sono stati trasferiti direttamente in ROM, come si può vedere dalla tavola riportata a pagina 42, e sono pertanto sempre disponibili. Molti comandi sono stati sensibilmente migliorati e quasi tutti sono stati riscritti *ex novo* in Assembly e i risultati, sia per quanto riguarda la velocità che per le dimensioni, si vedono. Tra le altre cose, sono stati aggiunti nuovi simboli nei template dei comandi oltre ai, si spera, conosciuti /A, /K e /S. Si tratta di /N che indica la presenza di un argomento numerico; /M che indica la possibilità d'inserire più di un argomento di quel tipo (per esempio la cancellazione di più file con il comando Delete); /F che indica un argomento stringa "finale", che dev'essere cioè l'ultimo al termine di una riga, e che può pertanto contenere anche degli spazi al suo interno: poiché dopo di esso non possono essere presenti altri parametri. Un esempio è la stringa da fornire al comando SetENV.

Alcuni comandi sono scomparsi del tutto, perché ormai obsoleti. Si tratta di EndCLI e di FF (*FastFonts*), il cui caratteristico aumento di velocità nella visualizzazione del testo fa ora parte integrante della graphics.library. L'unico comando veramente nuovo è invece MakeLink che apre un'intera gamma di nuove possibilità per l'utente e il programmatore. Questo comando consente infatti di avere sul disco un file che è in realtà il puntatore a un altro file, il quale, finalmente, contiene i dati veri e propri. Quando si tenta di accedere al primo file, i dati vengono letti, in maniera del tutto trasparente ai programmi e all'utente, dal secondo, che può trovarsi anche su un altro disco. Si può in tal modo ipotizzare, per esempio, un dischetto contenente un completo elenco di decine d'immagini grafiche, ma i cui dati risiedono, in realtà, su altri dischi, che verranno, quando necessario, richiesti automaticamente dal sistema. Una bella comodità, non c'è che dire!

Tra i comandi già conosciuti e semplicemente migliorati vale la pena di menzionare per lo meno Assign. In esso sono state aggiunte due caratteristiche che, e lo si può dire senza paura di esagerare, tutti gli utenti Amiga sognavano da anni. La prima è rappresentata dall'opzione DEFER. L'assegnare una directory con questo switch attivato fa sì che l'assegnazione stessa non abbia realmente luogo fino a quando non si fa esplicito riferimento alla directory in questione, richiedendo solo allora l'inserimento del dischetto corrispondente. Diventa così possibile assegnare una moltitudine di directory logiche direttamente nella startup-sequence senza bisogno di dover inserire subito tutti i dischetti necessari, come accadeva invece fino a oggi. Chi non possiede un hard disk, e magari nemmeno un secondo drive, non potrà invece non apprezzare l'opzione PATH. Oltre ad avere le stesse caratteristiche dell'opzione DEFER, l'assegnazione di una determinata directory logica come C: non viene più nemmeno vincolata a un particolare disco. Ecco un esempio che varrà a rendere tutto più chiaro: inserendo un disco "sconosciuto" in df0: e impartendo il comando Dir, non si aprirà più un requester del tipo "Insert volume Workbench1.3 in any drive", se il comando Dir è già presente nella directory principale o nella directory c del disco inserito, evitando così di dover ricorrere a scomodi trucchi del tipo "dir ?" solo per poter esaminare il contenuto di un dischetto.

A quando l'uscita?

Fino a ora, comunque, abbiamo a malapena sfiorato i cambiamenti più superficiali e appariscenti della versione 2.0, e rimandiamo l'ormai affaticato lettore al prossimo numero della rivista per un esame più "tecnico" di librerie, dispositivi e risorse, e delle loro novità e miglioramenti. Spendiamo invece qualche parola sul funzionamento e sulla reperibilità di questo nuovo *Kickstart*. L'unica versione momentaneamente disponibile al pubblico è quella, pressoché definitiva, fornita con l'Amiga 3000; per tutti gli altri modelli di Amiga esistono solo versioni "test", distribuite esclusivamente ai programmatori registrati presso la Commodore. Il *Kickstart* 2.0 non esiste infatti ancora sotto forma di ROM, ma solamente come file su disco. Questo file viene caricato, tramite un apposito programma chiamato KickIt, in una zona di RAM che viene così riservata al sistema. Dal momento che una caratteristica peculiare (almeno nell'ambiente Amiga) del *Kickstart* è quella di fare ampio uso al suo interno di indirizzamenti assoluti e non relativi, e di non essere pertanto rilocabile in memoria, correntemente esistono tre diverse versioni di questo file, denominate "f8", "20" e "c0". La prima, quella usata dall'Amiga 3000, deve venire caricata all'indirizzo $f80000. Si tratta dell'unica versione che sopravvive al reset ed è anche quella che potrebbe essere utilizzata sull'Amiga 1000... se solo l'Amiga 1000 avesse abbastanza spazio per contenerla, dal momento che la sua RAM "protetta" è di soli 256K. Alcune case produttrici di hardware hanno però già lanciato sul mercato schede in grado di raddoppiare questa quantità e di permettere inoltre il montaggio dei nuovi chip "ECS". Si tratta della Expert Services, con il suo "Rejuvenator" e della Delaware Valley Software che presenta invece il DVS-Wonder. Sarebbe bello poterle vedere anche in Italia.

Le versioni "20" e "c0" sono invece dedicate ad Amiga con espansione di memoria. La prima viene caricata all'indirizzo $200000 e per funzionare necessita quindi di un Amiga 500, 2000 o 2500 con una scheda configurata a questo indirizzo. Va bene anche la memoria fornita da una scheda acceleratrice come l'A2620 o A2630. La seconda versione viene invece usata su quei sistemi dotati di un solo megabyte di RAM e non provvisti del nuovo SuperAgnus poiché sfrutta la memoria presente a partire dall'indirizzo $C0000. Caratteristica comune a entrambe queste versioni è quella di dover essere ricaricate a ogni reset, rendendole perciò, specialmente per chi non possiede un hard disk in grado di abbreviare i tempi di caricamento (sono 512K!), di relativa utilità pratica, se non come curiosità o, appunto, a scopo di test.

La data entro la quale, invece, sarà resa disponibile al pubblico la versione definitiva dovrebbe teoricamente aggirarsi intorno al termine dell'anno in corso o ai primi mesi di quello venturo. Speriamo. ■

*Personalizza la tua maglietta !!
Se possiedi una stampante con i
nostri nastri lo puoi fare !!*

IN ARRIVO NUOVE CONSOLLE

DISCHETTI DA 3 1/2 DS-DD CERTIFICATI
L. 1000

DISPONIBILI ULTIME NOVITA' SOFTWARE & HARDWARE

ATARI LYNX

NEO-GEO

GAME BOY

SEGA MEGADRIVE

PC ENGINE & SUPERGRAPHIC

OFFERTA PC AT 286
1 Mb - 1 DRIVE HD - 1 HARD DISK DA 40 Mb
2 SERIALI - 1 PARALLELA - SCHEDA VGA 800 * 600
MONITOR VGA 1024 * 768 MONOC. - MOUSE
L. 1.590.000

NOVITA' SOFTWARE SETTIMANALI
PER AMIGA - C 64 - IBM

DRIVE INTERNO PER AMIGA 2000	L. 139.000
DRIVE ESTERNO PER AMIGA CON TRACK DISPLAY	L. 236.000
DRIVE ESTERNO PER AMIGA BYPASS	L. 179.000

DISPONIBILI HARD DISK PER AMIGA 500 - 1000 - 2000
DA 20 Mb A 63 Mb PER INFORMAZIONI TELEFONARE

ACTION REPLAY AMIGA L. 165.000

CASE IN METALLO PER AMIGA 500 L. 150.000

ESPANSIONI PER AMIGA

- 512 K INTERNA PER AMIGA 500	L. 140.000
- 2 Mb INTERNA PER AMIGA 500	L. 470.000
- 2 Mb ESTERNA PER AMIGA 500	L. 700.000
- 2 Mb ESTERNA PER AMIGA 1000 AUTOCON.	L. 800.000

PER AMIGA 2000 ESPANSIONI DA 4 Mb A 8 Mb L. TELEF.

TUTTI I PREZZI SONO IVA COMPRESA

GI. BIT COMPUTER VIA A. VISCONTI, 78 22053 LECCO (CO)
TEL. 0341/286241 - FAX IN LINEA - BANCA DATI 031/421391
031 COMPUTER VIALE MASIA 16/18 COMO

COMPUTER E TRASMISSIONE DATI

I PROTOCOLLI DI TRASFERIMENTO DALLA X ALLA Z

Dopo esserci occupati di XModem e YModem, iniziamo con questo articolo a esplorare lo ZMODEM, il protocollo oggi più in voga tra gli amanti di modem e BBS

di Avelino De Sabbata

Le parti precedenti di questa serie di articoli sono state pubblicate sui numeri 3-4/90.

Com'è stato sottolineato nel corso dei precedenti articoli, nei quali abbiamo analizzato il protocollo XModem e alcune delle sue estensioni, la creazione di Christensen iniziò a essere utilizzata da un'ampia varietà di sistemi, che ebbero così uno mezzo pratico, realizzabile senza difficoltà, col quale potersi scambiare agevolmente i dati. Tuttavia, pur con tutte le sue migliorie, XModem manteneva ancora limitazioni tali da essere considerato un protocollo adatto all'uso hobbistico. Allo scopo di non ripeterci, rimandiamo il lettore al precedente articolo nell'ambito del quale sono state esposte alcune limitazioni di XModem, che sono tra le principali motivazioni che spinsero la Telenet, all'inizio del 1986, a mettere in cantiere un progetto che aveva lo scopo di sviluppare e migliorare i protocolli e le applicazioni di pubblico dominio per il trasferimento di file. Le caratteristiche auspicate per il nuovo protocollo, poste quindi tra gli obiettivi di rilievo durante la progettazione, si concretizzarono nel programma Unix *rz/sz*. Questo programma, che venne realizzato adottando tutte le specifiche indicate dal nuovo protocollo, intanto battezzato ZModem, fu un ottimo compromesso ingegneristico tra diversi conflitti di esigenze.

Grazie all'impiego di una nuova tecnica, con la quale venne stabilito di usare particolari strutture predefinite grazie alle quali due terminali possono scambiarsi diverse informazioni oltre ai dati stessi, lo ZModem permette un'ampia serie di operazioni automatiche, tra le quali la possibilità per entrambi i programmi di comandare un trasferimento di file, d'inviare comandi all'altro programma o sistema operativo, di modificare alcuni parametri del trasferimento...

Con lo ZModem, i nomi dei file da trasferire vengono inseriti una volta sola da chi richiede il trasferimento, sia nel caso che la richiesta venga da parte dell'inviante sia che venga fatta dal ricevente, e grazie a ciò sono rese possibili le selezioni da menu e facili comandi per il controllo del protocollo, impartibili con la pressione di pochi tasti. All'interno del nome dei file possono essere utilizzati i caratteri jolly (wild card), con i quali diventa possibile trasferire una serie di file il cui nome risponde a particolari caratteristiche. Se il programma remoto non supporta ZModem, il protocollo è capace di abbassarsi con assoluta trasparenza al livello dell'YModem, ed effettuare correttamente il trasferimento.

Per quanto riguarda la produttività, ZModem avvia immediatamente il trasferimento, risparmiando i 10 secondi iniziali di temporizzazione necessari con XModem. Durante il trasferimento, vengono eliminate le pause per l'attesa del carattere di ritorno presenti in XModem, e i dati vengono inviati con continuità. Si tratta di una caratteristica già prevista per YModem-G, ma a differenza di YModem-G esiste la possibilità di recupero e correzione di eventuali errori intercettati durante la trasmissione, i quali invece fanno abortire la sessione YModem-G.

Tutti i trasferimenti ZModem sono protetti da un checksum CRC a 16 o 32 bit. Meccanismi di sicurezza sono inoltre previsti per proteggere il protocollo da finti messaggi di comando che potrebbero essere generati casualmente. La razionale generalizzazione del protocollo fa sì che possa essere adottato da un'ampia varietà di sistemi: elaboratori incapaci di gestire l'I/O seriale dei dati contemporaneamente alle operazioni sui dischi; elaboratori incapaci di gestire contemporaneamente trasmissione e ricezione sulla porta seriale; elaboratori e/o reti che richiedono un controllo del flusso con XON/XOFF; sistemi incapaci di verificare la presenza di dati nella coda di input.

Requisiti necessari

ZModem richiede un trasferimento dati a 8 bit, permettendo il trasferimento anche di pacchetti non codificati per ottenere la massima compatibilità con sistemi non del tutto trasparenti. ZModem ignora i caratteri di controllo del flusso, permettendo di operare con le linee a pacchetto commutate. In generale, opera al di sopra dei livelli supportati da XModem.

Al fine di sostenere un flusso dati pieno, controllato da XON-XOFF, oppure al di fuori dai controlli di flusso, come in X.25, gli accordi preliminari sostenuti dal protocollo dovrebbero indicare il controllo di flusso da attuare, oppure la possibilità di trasmissione a piena velocità senza perdita di dati. Negli altri casi l'inviante deve tener conto dell'ampiezza della window.

ZModem non pone alcuna limitazione sulle informazioni contenute nei file binari, eccetto per il numero di bit del file che dev'essere un multiplo di otto. Com'è gia stato posto all'attenzione dei lettori in un precedente articolo, dal momento che ZModem viene usato per trasferire file tra differenti sistemi operativi, i file di testo debbono concordare su un minimo di requisiti per poter essere leggibili su un'ampia varietà di sistemi. Le linee di testo consistono in caratteri ASCII stampabili, spazi, tab e backspace. Per quanto riguarda le convenzioni da adottare sui caratteri terminatori di linea EOL, si vedano gli articoli precedenti. L'inviante ZModem comunque, grazie alle opzioni di conversione (flag ZCNL), può forzare il ricevente a convertire il fine linea ricevuto nel suo EOL convenzionale.

ZModem ottiene la trasparenza dei dati estendendo il set di caratteri a 8 bit (256 codici) con una sequenza di escape basata sul carattere ZDLE (ZModem Data Link Escape). Questo codice permette subpacchetti di dati di lunghezza variabile senza per questo appesantire il flusso dei dati con byte per il conteggio dei caratteri. Inoltre, permette di verificare l'inizio della struttura rinunciando alle dispersive tecniche di temporizzazione, rendendo in questo modo estremamente veloce l'attivazione delle procedure di recupero degli errori. Il carattere ZDLE è un carattere speciale, che in alcuni casi rappresenta un carattere di controllo.

Il valore ZDLE (definito nel file "zmodem.h") è l'ASCII $18, CAN. Questo valore è stato adottato per consentire l'abbandono della sessione ZModem al ricevimento di 5 caratteri CAN consecutivi, il che garantisce la compatibilità con i protocolli inferiori. Nonostante il protocollo indichi come sequenza di abbandono 5 caratteri CAN, normalmente viene attivata con l'invio di almeno 8 di questi caratteri.

Dal momento che CAN non viene utilizzato durante le normali operazioni di trasferimento, applicazioni interattive e programmi di comunicazione possono controllare il flusso dei dati grazie all'uso di ZDLE.

Se il valore ZDLE compare in un file binario, l'inviante avrà cura di prefissarlo con un ulteriore carattere ZDLE, e quindi inviarlo come sequenza ZDLEE. Il programma ricevente decodifica qualsiasi sequenza ZDLE seguita da un byte con il bit 6 a zero e il bit cinque a uno (con qualsiasi parità), nel corrispondente carattere di controllo grazie all'inversione del bit 6 operata sul carattere ricevuto. Le sequenze di controllo permesso sono pertanto determinate dai caratteri ASCII $40-$5F e da tutto il set esteso ASCII $7F-$FF.

Le routine "zm.c" accettano un'opzione di escape per ognuno dei caratteri di controllo disponibili, rendendo possibili operazioni con reti con minor trasparenza. Queste opzioni possono essere richieste nella fase iniziale dai programmi ricevente e trasmittente.

Studio ed evoluzione dello ZModem

ZModem fu sviluppato per il pubblico dominio, sotto un contratto con la Telenet. Le descrizioni del protocollo e il codice del programma sorgente *rz/sz*, al quale è auspicabile ispirarsi per produrre conversioni completamente compatibili, sono di pubblico dominio. Nessuna restrizione per copyright è applicata sull'uso del protocollo, del quale al contrario viene incoraggiata la diffusione e l'uso su nuovi sistemi.

All'inizio si pensò che poche modifiche all'XModem avrebbero permesso un notevole miglioramento nelle prestazioni sulle reti a pacchetto commutate, mantenendo allo stesso tempo la semplicità del protocollo originale. L'idea di base da cui ebbe inizio lo studio era di contrassegnare con una numerazione sequenziale i caratteri ACK o NAK, o di creare dei blocchi numerati a cui affidare la risposta positiva o negativa di ritorno. Ciò avrebbe permesso all'inviante di spedire più blocchi di dati uno di seguito all'altro, senza doversi fermare ad attendere una risposta per ognuno.

Il problema però era sul modo di aggiungere questo blocco numerato agli ACK e NAK di ritorno. Altri protocolli (SEAlink, MEGAlink, WXModem), aggiungono dei byte binari per indicare il numero di blocco. Questo sistema però non poteva essere usato in ZModem, in quanto alcune combinazioni di codici binari non sarebbero passate in modo bidirezionale sia in certi modem che in particolari reti e sistemi operativi. Altri sistemi operativi avrebbero potuto non riconoscere caratteri di ritorno senza fermarsi in attesa, a meno che non fossero presenti altri particolari segnali d'interruzione o dipendenti dal proprio codice... Questo problema venne risolto trasformando i semplici caratteri ACK e NAK in veri e propri pacchetti.

La gestione delle WINDOW fu un altro problema. In questo caso con l'espressione WINDOW si vuole indicare la quantità di caratteri inviati dal trasmittente e non ancora ricevuti dal ricevente. Esperienze precedenti, acquisite nella programmazione del protocollo SuperKermit davano per necessaria alla velocità di 1200 bps, un'ampiezza della window di almeno 1000 caratteri. I nuovi modem ad alta velocità, per funzionare con il massimo rendimento, richiedono ora una window di 20000 o più caratteri.

Un punto dolente con XModem e la sua progenie è il recupero degli errori. Come può il ricevente determinare comunque se l'inviante ha risposto, oppure se è pronto a rispondere a una richiesta di ritrasmissione di dati mal ricevuti? XModem affronta il problema ignorando qualsiasi carattere in arrivo fino a che rileva un determinato periodo di silenzio. La determinazione della lunghezza della pausa è però fonte di errate interpretazioni; un periodo troppo breve infatti permette a una pausa spuria in arrivo di essere considerata un recupero d'errore. Un "timeout" troppo lungo d'altro canto produce danni ancora maggiori, permettendo a particolari disturbi di linea di agganciarsi al protocollo. SuperKermit risolve il problema con un distinto carattere d'inizio pacchetto (SOH). W-XModem e ZModem adottano una sequenza di caratteri unica e inequivocabile per determinare l'inizio di una struttura. Sealink e Megalink non si curano del problema.

Altro difetto di XModem e W-XModem è il "padding", ovvero i caratteri estranei al file che vengono aggiunti alla fine dell'ultimo pacchetto dati. Se questo poteva risultare accettabile con vecchi sistemi operativi come il CP/M, dove l'esatta lunghezza dei file non era determinante, non lo è più con sistemi come MS-DOS, UNIX... Anche se il problema veniva parzialmente

HACKER'S HANDBOOK III

(di Hugo Cornwall, Century Hutchinson, Londra 1983, pag 236, £. 6,95, ISBN 0-7126-1147-9)

Susan Headley aveva soltanto 17 anni quando si mise nei guai. Non nel genere di guai a cui probabilmente state pensando in questo momento: tutt'altro. Il fatto è che col proprio computer era riuscita a entrare nelle centrali della compagnia telefonica e, attraverso quelle, a scorrazzare per l'intera rete a piacimento e per diverso tempo.

Nel 1982 Ronald Austin e il suo amico Kevin Poulsen erano due studenti sui vent'anni. Con un semplice TRS80 e un Commodore Vic 20 smentirono clamorosamente il noto detto secondo cui la curiosità è donna. Lo fecero andando a mettere di proposito il naso nel computer del NORAD (sì, proprio quello di *War Games*). Non che ci siano arrivati subito, intendiamoci: prima avevano fatto alcuni "giretti" praticamente per tutti gli USA, cominciando dal CED dell'Università di Berkeley, passando per quello di Stanford, poi andando a quello di un altro importante centro di Washington, con perfino una puntatina all'agenzia norvegese di telecomunicazioni. Il tutto usufruendo dell'Advanced Research Projects Agency network (ARPAnet). Erano riusciti a entrare attraverso l'Università di Berkeley e non si resero conto che quello porta sarebbe stata loro fatale a lungo andare poiché, sebbene quell'Università facesse parte dell'ARPAnet, non usava molto questa rete, così qualcuno, vedendo un frequente e come tale insolito movimento dal CED di Berkeley, s'insospettì e per Ron e Kevin fu l'inizio della fine.

Ma c'è stato chi ha saputo combinare un pandemonio ancora più grande. Tra la primavera e l'autunno dell'84 quattro ragazzotti londinesi pensarono di fare al PRESTEL, il servizio Videotex Britannico, uno di quegli scherzi in grande stile alla *Amici miei*. Lo scherzo consistette semplicemente nel sostituirsi pari pari al o ai sysop di quel servizio (non pagando naturalmente alcun addebito oltre alla telefonata). Potevano inventare e assegnare password oppure cancellarne altre; editare fantasiosamente pagine oppure sbarazzarsene. Il tutto quando e come avessero voluto. Furono scoperti perché, in fondo, per vanità, si lasciarono scoprire.

Infatti uno dei quattro, tale Robert Shifreen, a un certo punto fece una dimostrazione con alle spalle addirittura un giornalista come testimone (e chi ha detto che la vanità è donna?). Detto questo, è facile immaginare di cosa parlasse un certo giornale il giorno dopo e il putiferio che conseguentemente scatenò per parecchie settimane a venire. Non vi dico più niente; vi consiglio soltanto di procurarvi in qualche modo il libro di Hugo Cornwall *Hacker's Handbook III* e andarvi a leggere (in inglese, il libro non si trova in Italia né è stato tradotto) a pagina 143 e dintorni com'è andata a finire la storia. E chi non sa l'inglese? Può fare due cose: o impararselo, o ritagliarsi questa recensione e tenersela per ricordo. A parte gli scherzi, il libro di Cornwall, giornalista e hacker lui stesso, è veramente ben fatto e piacevolissimo da leggere. È un miscuglio tra il manuale tecnico, molto ben curato e dettagliato, e la spy-story. Non mancano nemmeno consigli che potrebbero tornare utili anche a chi pensa di essere l'hacker più esperto e crede (a torto) di non aver più nulla da imparare. Dopo poche pagine, si capisce che per l'autore, l'hacker – oltre a non essere, come spesso si potrebbe credere, un ladro – non è certo quello che, avendo avuto la buona dritta di un numero e una password, comincia a curiosare qua e là in una banca dati: un hacker è ben altro. Non è soltanto questione di provare a indovinare password a furia di tentativi: spesso le password (o indizi che possono condurre alla loro scoperta) sono quasi sotto il sole. Basta saper dove guardare: anche nei cestini della carta straccia di certe aziende. E stare pazientemente in agguato in attesa di una buona occasione. Anche una buona conoscenza della macchina su cui gira il network da scardinare, nonché del suo sistema operativo sono necessari. E in questo suo libro Cornwall li esamina e descrive un po' tutti: quali sono le loro caratteristiche peculiari cosa possono offrire le grandi banche dati che funzionano sui mainframe, le loro protezioni più usuali, cosa fare una volta trovata la PW e tante altre notizie interessanti. Non tralascia nemmeno l'hacking via etere ovvero quello che vede protagonisti la radio e l'elaboratore e la TV col Televideo o consimili. È tutto e spiegato partendo dall'HW e il SW di volta in volta occorrenti: quanto costa, dove si può trovarlo e come usarlo. Naturalmente, è sempre da tener presente che il libro si rivolge ad un pubblico anglosassone e in particolar modo britannico. Ciò non significa che la sua lettura non possa dare spunti e "illuminazioni" anche agli hacker nostrani; né suscitare una certa curiosità "sociologica" in chi hacker non è né – come chi scrive – ha alcuna intenzione di diventarlo.

S.F.

risolto in YModem il quale invia assieme ad altre informazioni anche la lunghezza del file nell'header, una soluzione più efficiente si rivela l'adozione di pacchetti di lunghezza variabile, come venne adottato dal Kermit.

Un ulteriore problema nei protocolli di pubblico dominio che hanno preceduto ZModem è la difficoltà del ricevente di accertarsi se è stato riconosciuto il proprio messaggio di errore. Fino a completo ricevimento del pacchetto in arrivo infatti, non è dato di sapere se questo è la corretta risposta alla richiesta di ritrasmissione inviata oppure se deve essere ignorato perché l'inviante non ha riconosciuto il segnale di errore. Inoltre, quanto dovrà continuare questo stato di cose prima che il ricevente si preoccupi d'inviare un'ulteriore messaggio di errore?

SuperKermit, a costo di un notevole incremento della complessità porta sensibili miglioramenti al rendimento del protocollo Kermit, di cui è una estensione, usa ritrasmissioni selettive, in modo da poter accettare qualsiasi pacchetto ricevuto correttamente. I pacchetti ricevuti con errori vengono "annotati" in una lista, e ogni volta che SuperKermit riceve un pacchetto dati, deve decidere quale dei pacchetti in sospeso, sempre che ve ne siano, vuole ricevere per primo, inviando una specifica richiesta in tal senso al trasmittente. Procedure con le caratteristiche sopra accennate, richiedono chiaramente la realizzazione di software molto complesso per ottenere un'accettabile livello di affidabilità. Per alcuni dettagli su Kermit si veda la parte dell'articolo dedicata a un approfondimento dei principi di questo protocollo.

Per la realizzazione di ZModem fu deciso di dimenticare la complessità del SuperKermit, il quale, in relazione al suo efficiente ma complicato schema di gestione dei pacchetti, richiede tra l'altro una notevole quantità di memoria da mettere a disposizione per il buffer.

Dal momento che alcuni caratteri possono essere persi o corrotti in qualsiasi punto, non esiste la possibilità d'inserire in una determinata posizione eventuali caratteri non ricevuti (il che perlomeno eviterebbe la perdita del sincronismo), o di determinare la posizione in cui specificare di volta in volta, la lunghezza (variabile) del pacchetto. In ZModem la lunghezza di ogni subpacchetto dati è determinata dal terminare ciascun subpacchetto da una sequenza di escape simile a quelle adottate dal Bisync e HDLC. Il risultato finale è l'header ZMOEM, contenente una "frame type", ossia una struttura che definisce la natura di ciò che segue.

ZMODEM e gli header

A questo punto è necessario sottolineare che nel corso dell'articolo, piuttosto che riferirci alle costanti standardizzate nel protocollo con sterili valori numerici, le indicheremo con i nomi che furono a queste assegnati dalla Telenet al momento della stesura degli algoritmi per il programma Unix *rz/sz*. Gli assegnamenti dei rispettivi valori avvengono nel file zmodem.h che fa parte dei file di compilazione del programma *rz/sz*. In questo stesso file, reperibile con i sorgenti del programma nei BBS più forniti, i lettori più esigenti potranno rilevare anche quei valori non riportati nel corso dell'articolo, che non dovrà essere ritenuto esaustivo su quanto trattato, a causa della notevole complessità

dell'argomento.

Per comprendere la caratteristica innovatrice del protocollo ZModem è necessario abbandonare lo schema mentale legato al modello X-YModem, con il quale i dati viaggiano unidirezionalmente in pacchetti da 128 o da 1024 byte, la cui corretta o errata ricezione viene comunicata rispettivamente dai caratteri di ritorno ACK e NAK. Nello ZModem, questi caratteri sono evoluti in veri e propri pacchetti. Esistono diversi tipi di pacchetti, che in ZModem vengono definiti FRAME (struttura) con funzioni differenti a seconda delle varie esigenze.

Tutte le strutture iniziano con un HEADER che viene inviato in forma binaria oppure in esadecimale. Il trasferimento dei dati avviene in DATA FRAME (strutture dati), costituite di un header e seguite da uno o più subpacchetti dati. In assenza di errori, un intero file può essere trasmesso in un unico data frame.

Entrambe le forme di header (binaria o esadecimale) contengono le stesse informazioni: viene indicato il tipo di struttura con il primo byte, e nei successivi quattro byte vengono riportati alcuni flag e/o valori dipendenti dal tipo stesso della struttura. Un tipico header assume uno dei due seguenti schemi:

```
TYPE  ZF3  ZF2  ZF1  ZF0

TYPE  ZP0  ZP1  ZP2  ZP3
```

dove:

```
TYPE       indica il tipo di struttura
ZF0-ZF3    indicano i flag usati per diversi scopi
            (ZF0 byte meno significativo)
ZP0-ZP3     File Position, indicano i byte usati
           per indicare l'offset del file
           (ZP0 byte meno significativo)
```

I byte dell'header, (ZFx / ZPx), eventualmente non utilizzati, devono essere sempre impostati a zero.

Anche gli header però sono di fatto considerati strutture (frame type), che definiscono il tipo di struttura che precedono. Infatti oltre ai cinque byte di cui si è parlato più sopra, una "frame type" è preceduta e seguita da altri caratteri che la completano e che vengono di seguito analizzati.

Esistono tre tipi diversi di header: binario con checksum a 16 bit, binario con crc a 32 bit e infine header esadecimale. In ZModem esiste un'unica routine "zgethdr", la quale è capace di riconoscere tutti i tipi di header. La funzione "zsbhdr" invia un header binario con CRC sia a 16 che a 32 bit, mentre la funzione "zshhdr" invia un header esadecimale. Qualsiasi header inizia con il carattere ZPAD, ("*" ASCII $2A).

L'header binario viene inviato dal programma trasmittente al ricevente e tutti i byte componenti la struttura "frame type" sono codificati con il codice di escape ZDLE a eccezione del primo, (ZPAD). La riga che segue rappresenta la struttura di un header binario con checksum a 16 bit, dove ZBIN è il carattere "A" (ASCII $41).

```
ZPAD ZBIN TYPE F3/P0 F2/P1 F1/P2 F0/P3 CRC1 CRC2
```

Lo schema che segue rappresenta invece la struttura di un header binario con checksum a 32 bit che è uguale alla precedente, tranne che per la costante ZBIN che è sostituita da ZBIN32 (carattere "C", ASCII $43), e per i byte di checksum che ora sono quattro.

```
ZPAD ZBIN32 TYPE F3/P0 F2/P1 F1/P2 F0/P3 CRC1 CRC2
CRC3 CRC4
```

In relazione al tipo di struttura annunciata dall'header, zero o più subpacchetti di dati binari (con CRC rispettivamente a 16 o a 32 bit), potranno far seguito alla struttura "frame type".

Le strutture con header esadecimali vengono usate dal ricevente per inviare le proprie risposte. Gli header esadecimali sono inoltre usati anche dall'inviante quando questi non siano seguiti da subpacchetti di dati binari. L'uso di header in esadecimale da parte del ricevente permette un agevole uso dei caratteri di controllo per interrompere il flusso dei dati nel momento in cui viene rilevato un errore.

Sia l'header esadecimale stesso, sia i subpacchetti dati che a esso dovessero seguire, vengono sempre protetti con due soli byte di checksum. Visto che questo tipo di codifica ha la peculiarità di rilevare molti degli errori che dovessero verificarsi durante il trasferimento, un header esadecimale con checksum a 32 bit, per quel che ne sappiamo, non è stato ancora definito.

L'header in esadecimale inizia con la sequenza di caratteri ZPAD ZPAD ZDLE ZHEX. La routine "zgethdr" si sincronizza con la sequenza ZPAD ZDLE. Il carattere ZPAD in più permette al programma inviante di verificare un header asincrono che indicherebbe una condizione d'errore. La costante ZHEX è rappresentata dal carattere "B" (ASCII $42). Tutti i byte della struttura sono inviati in esadecimale usando il set di caratteri 0123456789abcdef, e i caratteri "abcdef" devono far parte del set minuscolo (ASCII $61-$66).

Una particolarità rilevante è che le strutture codificate in esadecimale devono terminare con un normale EOL (End Of Line), costituito dai caratteri CR/LF. La routine di ricezione si aspetta di trovare almeno uno di questi caratteri, due nel caso che il primo ricevuto sia CR. Inoltre, alla fine di ogni struttura esadecimale viene accodato un carattere XON, a eccezione delle strutture ZACK e ZFIN. Il carattere XON ristabilisce il normale flusso nel caso il ricevente intercetti un indebito XOFF, generato casualmente da disturbi di linea.

XON non viene inviato dopo una struttura ZACK per proteggere eventuali situazioni a flusso controllato, mentre dopo una struttura ZFIN non viene inviato per permettere la corretta esecuzione della fase di pulizia della linea (PURGE) che chiude la sessione ZModem.

Lo schema che segue rappresenta quanto detto sull'header esadecimale. Si tenga presente che dal byte TYPE fino ai due byte CRC compresi (sottolineati nello schema che segue), i caratteri vengono inviati sotto forma di valori esadecimali.

```
ZPAD ZPAD ZDLE ZHEX TYPE F3/P0 F2/P1 F1/P2 F0/P3 CRC1
CRC2 CR LF XON
```

I subpacchetti di dati binari che possono contenere da zero a 1024 byte, seguono immediatamente il pacchetto header binario. La lunghezza dei subpacchetti di dati dovrebbe essere proporzionale alla velocità del trasferimento in atto. Il protocollo raccomanda che al di sotto dei 2400 bps la lunghezza non sia superiore a 256 byte, fra i 2400 e i 4800 bps sia di 512 byte e al di sopra dei 4800 bps, oppure quando il collegamento si rivela sufficentemente esente da errori, questa raggiunga i 1024 byte. I byte dei dati prima di essere inviati sono

codificati ricorrendo al carattere ZDLE. Il termine viene segnalato da uno ZDLE e da una struttura frameend che segnala la fine del file, seguita da due o da quattro byte di checksum anche questi codificati con ZDLE. Nel calcolo del CRC intervengono tutti i byte di dati e la struttura "frameend". Le funzioni preposte all'invio e alla ricezione dei subpacchetti di dati nel programma *rz/sz* sono rispettivamente "zsdata" e "zrdata". Per i trasferimenti di subpacchetti di dati in ASCII, usati principalmente in ambienti a sette bit, correntemente non è definito un formato di codifica.

ZRQINIT: questa struttura viene trasmessa dal programma inviante, per segnalare al programma ricevente l'inizio dell'elaborazione, invitandolo a inviare il suo header ZRINIT. Questo evita l'inutile ritardo iniziale che si ha con le temporizzazioni associate ai trasferimenti con Kermit e XModem. Il programma inviante può ripetere l'invito di ricezione (trasmettendo ancora ZRQINIT) fino a che non ottiene risposta. Se il programma sta tentando d'inviare un comando, ZF0 viene opportunamente settato al valore corrispondente, (ZCOMMAND), altrimenti è posto a zero.

ZRINIT: viene inviata in risposta a ZRQINIT dal programma ricevente. ZF0 e ZF1 contengono la modalità di ricezione oppure dei flag che indicano alcune capacità del ricevente secondo la tabella che segue. ZP0 e ZP1 contengono l'ampiezza del buffer del ricevente in byte, oppure zero se è permesso l'I/O senza interruzione del flusso dati (flag CANOVIO).

[illegible]

Le strutture

ZSINIT: l'inviante invia un subpacchetto di dati binari terminanti con una struttura ZCRCW. Il byte ZF0 dell'header è mascherato con i flag della tavola riportata più sotto. Il subpacchetto dati contiene la sequenza Attn terminante con un byte nullo, della lunghezza massima di 32 byte, incluso il byte nullo.

[illegible]

ZACK: riconoscimento della struttura ZSINIT, dell'header ZSCHALLENGE, e dei subpacchetti dati ZCRCQ o ZCRCW. In ZP0-ZP3 è contenuto l'offset del file. Nella risposta a ZCHALLENGE in ZP0-ZP3 viene riportato lo stesso valore a 32 bit ricevuto nell'header ZCHALLENGE.

ZSKIP: questa struttura quando viene utilizzata dal ricevente in risposta a ZFILE, ordina al trasmittente d'ignorare il trasferimento del file corrente e passare subito alla trasmissione del file successivo.

ZNAK: analogamente al segnale NAK di XModem, la struttura ZNAK indica il rilevamento di errori nel ricevimento dell'ultimo header (si veda anche ZRPOS).

ZABORT: viene inviata dal ricevente per porre termine al trasferimento di file batch quando l'interruzione è richiesta dall'utente. L'inviante risponde con una sequenza ZFIN.

ZFIN: la struttura ZFIN viene trasmessa dal programma inviante per terminare una sessione ZMODEM. Il ricevente risponde con una propria struttura ZFIN.

ZRPOS: questa struttura viene inviata dal ricevente al determinarsi di un errore, per forzare il trasmittente a riprendere il trasferimento a un determinato offset indicato dal puntatore ZP0-ZP3.

ZDATA: questa è la vera e propria struttura utilizzata per l'invio dei dati. ZP0-ZP3 contengono l'offset file. Possono seguire uno o più subpacchetti dati.

ZEOF: l'inviante segnala la fine del file, inviando in ZP0-ZP3 l'offset del carattere terminatore del file EOF.

ZFERR: segnala un errore in lettura o in scrittura del file, che non è recuperabile, il che equivale all'invio di una struttura ZABORT.

ZCRC: struttura di richiesta (da parte del ricevente) e di risposta (da parte dell'inviante) di codici di controllo polinomiali (CRC), che vengono inviati in ZP0-ZP3.

ZCHALLENGE: con questa struttura il ricevente chiede che gli venga inviata una struttura ZACK contenente un numero casuale in ZP0-ZP3. Questo permette al programma ricevente di verificare che la connessione sia effettivamente attivata con un programma operativo, e che non sia stata invece accidentalmente attivata da interferenze di linea.

ZCOMPL: con questa struttura si indica la conclusione dell'ultima operazione richiesta.

ZCAN: questa è una pseudo-struttura che viene reinviata dalla funzione gethdr() in risposta a una sequenza di abort.

ZFREECNT: struttura utilizzata dal programma inviante tramite la quale richiede di ritorno una struttura ZACK la quale riporta in ZP0-ZP3 il numero dei byte a disposizione del file system ricevente. Il valore zero rappresenta una quantità di byte non definita.

ZFILE: questa struttura rivela l'inizio di un tentativo di trasmissione dati, i flag.

ZF0-ZF3 sono usati per accordare trasmittente e ricevente su alcune opzioni che lo ZModem rende possibili durante il trasferimento. In ZF0 vengono riportate le opzioni di conversione, in ZF1 le opzioni di gestione, in ZF2 le opzioni di trasporto, mentre il byte ZF3 è dedicato alle opzioni estese. Se questi flag sono resettati (valore zero), significa che non viene richiesto alcun trattamento particolare. Sempre in accordo con il

fatto di essere un protocollo "receiven driver", le opzioni specificate dal ricevente prevalgono sulle opzioni indicate dall'inviante con l'eccezione di ZCBIN che annulla qualsiasi altra opzione di conversione eventualmente richiesta sia dall'inviante che dal ricevente.

Opzioni di conversione (Flag ZF0). Se il ricevente non riconosce le opzioni di conversione, può essere prevista una conversione di default che dipenderà dal sistema operativo su cui è implementato il protocollo.

ZCBIN indica che il trasferimento avviene in modo binario, e pertanto la presenza di questo flag inibisce incondizionatamente qualsiasi altra conversione.

ZCNL converte il/i caratteri terminatori di linea ricevuti con i codici EOF supportati dal sistema in cui avviene la ricezione. I terminatori di linea convenzionali sono la coppia di caratteri CR/LF usati in molti dei sistemi basati sul codice ASCII eccetto Unix, AmigaDOS e Macintosh. In questi ultimi sistemi operativi viene usato il carattere NL (NewLine). Entrambe questi terminatori di linea accettano le definizioni permissive dell'ASCII per il ritorno carrello e Line Feed-New Line, mentre linee separate solamente da ritorni carrello non sono contemplati né dal codice ASCII né tantomeno da ZCNL in ZModem.

ZCRECOV indica la capacità dell'implementazione di riprendere un trasferimento accidentalmente interrotto dal punto in cui questo era giunto nel momento in cui è avvenuta l'interruzione. Questa opzione è utile per aggiornare la copia di un file senza reinviare completamente i vecchi dati, a patto che la nuova versione del file non sia più corta dell'esistente. Se il file di destinazione esiste e non è più lungo del file sorgente, si può effettuare un append con il file di destinazione e iniziare il trasferimento all'offset corrispondente al termine del file esistente. File che abbiano subito una conversione, (flag ZCNL), o che siano stati soggetti a opzioni di trasporto, non possono usufruire del Transfer Recovery.

Opzioni di gestione (Flag ZF1). Se il ricevente non riconosce le opzioni di gestione, il file verrà trasferito normalmente.

Il bit ZMSKNOLOC dice al ricevente di passare il file corrente solo se il ricevente non ha un file con lo stesso nome. Cinque bit (definiti da ZMMASK) definiscono il seguente set di opzioni di gestione le quali sono fra di loro mutualmente esclusive.

ZMNEWL abilita il trasferimento solo se il file destinazione non esiste, altrimenti viene controllata la data e la lunghezza del file esistente, e il trasferimento avviene solo se il file da trasferire è più recente o più lungo di quello esistente.

ZMCRC compara il file sorgente con il file destinazione ed effettua il trasferimento solo se la lunghezza del file o se il calcolo del CRC polinomiale non coincidono.

ZMAPND appende i contenuti del file sorgente alla fine del file destinazione sempre che esista.

ZMCLOB rimpiazza il file destinazione esistente se c'è.

ZMDIFF trasferisce il file se il file destinazione è assente. Altrimenti compara la data e la lunghezza dei file sorgente e destinazione, effettuando il trasferimento se vengono riscontrate differenze.

ZMPROT crea una protezione del file destinazione, in quanto il trasferimento viene effettuato solo se questo non esiste.

ZMNEW effettua il trasferimento se il file destinazione è assente. Altrimenti, controlla se la data del file sorgente è più recente, e solo in caso affermativo effettua il trasferimento.

Opzioni di trasporto (Flag ZF2). Se il ricevente non implementa la particolare opzione di trasporto, il file viene copiato senza alcuna conversione.

ZTLZW abilita la compressione con il metodo Lempel-Ziv. I dati trasmessi saranno identici a quelli prodotti con VAX byte ordering 4.0, usando una codifica a 12 bit.

ZTCRYPT abilita la crittazione del file trasmesso. Una stringa iniziale terminante con un carattere NULL identifica la chiave di crittazione.

ZTRLE esegue una codifica in lunghezza. Di seguito viene inviato un subpacchetto dati ZCRCW, con il quale viene inviato il nome del file, la sua lunghezza, la data di modifica, e altre informazioni che vengono descritte in altra parte dell'articolo.

Opzioni estese (Flag ZF3). Il valore indicato dalla costante ZTSPARS abilita uno speciale modo di elaborazione che permette il trasferimento di segmenti di file sparsi, o ritrasmissioni selettive che vengono gestite dall'inviante. Ciascun segmento del file viene trasmesso come una struttura indipendente, e le varie strutture trasferite possono anche comprendere "brani" non contigui del file. L'inviante dovrebbe terminare ciascun segmento con un subpacchetto dati ZCRCW (che implica l'invio di una risposta), e attendere quindi la conferma di ricevimento da una struttura ZACK al fine di assicurarsi che nessun dato vada perso. La procedura ZTSPARS non può però essere utilizzata se è attiva l'opzione di conversione ZCNL.

ZCOMMAND: la struttura ZCOMMAND viene inviata in una struttura binaria nella quale il flag ZF0 contiene zero oppure il valore ZCACK1. Segue un subpacchetto dati ZCRCW, con la stringa di comando sotto forma di testo in caratteri ASCII terminante con un carattere nullo. Se il comando dev'essere eseguito dal sistema operativo che ospita il programma ricevente il nome del comando deve avere come primo carattere "!". Altrimenti s'intende debba essere eseguito dal programma applicativo che riceve il comando stesso.

Se il ricevente rivela l'arrivo di un comando illegale o dal formato errato, risponde immediatamente con un header ZCOMPL riportando il codice dell'errore riscontrato in ZP0-ZP3. Se ZF0 contiene il valore ZCACK1, il ricevente risponde con un header ZCOMPL in cui ZP0-ZP3 sono resettati (posti a zero), altrimenti l'header ZCOMPL viene inviata solo quando l'operazione è completata. Il codice di errore di ritorno dal programma che riceve il comando viene immagazzinato in ZP0-ZP3. Un valore zero in uscita indica il successo e il completamento del comando. Se il comando formulato richiede la trasmissione di un file, il programma che ha trasmesso il comando attende l'arrivo di una struttura ZRQINIT dal computer collegato, la quale segnala l'inizio del trasferimento dati. Se il programma ricevente non implementa la possibilità di attivare il download automatico, questi deve visualizzare la richiesta ricevuta sullo schermo del proprio terminale e poi inviare un header ZCOMPL anche se effettivamente il comando non è stato eseguito. L'inviante esamina il flag ZF0 dell'header ZRQINIT ricevuto per accertarsi che non sia un'eco del proprio header ZRQINIT. Non è ammesso che il programma inviante comandi al programma ricevente d'inviare un comando. ■

UNO SGUARDO SUGLI USA

QUI STATI UNITI D'AMERICA

In questo numero: Scholastic Software, scanner manuale Migraph, Professional Draw 2.0, Excellence! 2.0, ProWrite 3.0, Art Department della ASDG e altro ancora.

di Morton A. Kevelson

Il numero di agosto della rivista *Computer Shopper*, nella sua sezione dedicata al mondo IBM/MS-DOS, parlava con enfasi di "programmi di CAD che funzionano in ambiente DOS, Windows e anche OS/2". Per sottolineare il tutto i grafici della rivista avevano messo in bella mostra nell'indice una schermata dall'aspetto molto professionale di un CAD che mostrava delle visioni laterali di un complesso congegno meccanico. L'aria di professionalità della schermata era sottolineata dalle potenti funzioni messe a disposizione da una fila di settanta icone che correvano su un lato e nella parte superiore della schermata. Queste icone rendevano senza dubbio il programma molto potente e di facile uso. La stessa identica fotografia la si ritrovava però anche a pagina 613 nell'articolo di Peggy Herrington "Amiga one stop". Sembra infatti che la schermata appartenga a *Dyna-CADD*, un nuovo programma di CAD per l'Amiga che viene venduto a 995 dollari!

Un breve esame all'articolo sui CAD per MS-DOS rivelò poi che le foto relative alle schermate dei CAD basati sui computer MS-DOS, non erano neanche lontanamente d'impatto paragonabile a quella di *Dyna-CADD*. Apparentemente, il grafico di *Computer Shopper*, quando si è trovato dinanzi a una serie di foto di schermate di programmi di CAD, ha scelto per l'indice quella dall'aspetto più professionale. La mia esperienza personale mi dice che gli schermi di lavoro dei CAD per l'Amiga stanno cominciando a richiamare da vicino quelli delle workstation che vengono utilizzate per illustrare le pubblicazioni d'ingegneria.

Compute! colpisce ancora

Gli abbonati a *Compute!*, o quelli a una delle sue più specifiche pubblicazioni quali *Compute!'s Gazette* e *Compute!'s Amiga Resource*, hanno ricevuto una copia della rivista *OMNI* di agosto. *OMNI* aveva al suo interno una speciale sezione di *Compute* (da notare che il punto esclamativo è stato eliminato).

Novità didattica

La Scholastic Software, in collaborazione con la Commodore USA, sta sviluppando un prodotto didattico multimediale e interattivo basato sull'Amiga 500. Il package sarà costituito da un Amiga 500 accoppiato a un lettore di videodischi da 12 pollici. Il primo software per questo sistema dovrebbe chiamarsi *Visions of American History*. Il soggetto riguarda la storia del movimento per i diritti civili e riguarda gli indiani, gli uomini di colore e gli ispanoamericani.

La Scholastic Software ha presentato il tutto al meeting AMUSE svoltosi a New York a metà luglio. Il programma, che è basato sul nuovo sistema autore *AmigaVision*, funzionava bene e la gestione dei soggetti risultava ben organizzata. Sulla base di quanto ho appreso dai programmatori, uno degli aspetti di maggior difficoltà del progetto è stato il reperire e ottenere i diritti di pubblicazione dei vari filmati storici e inserti sonori che riempiono larga parte del videodisco. Basandomi su quanto ho visto alla presentazione, ritengo che questo prodotto vada ben oltre quello che gli insegnanti avevano in mente una decina d'anni fa quando furono introdotti per la prima volta i computer nelle classi.

Nuovi arrivi e aggiornamenti

Finalmente è arrivata la versione 1.0 del software *Touch-Up* per lo scanner manuale Migraph, seguita a una sola settimana di distanza dalla 1.01. Gli addetti alle pubbliche relazioni della Migraph non erano probabilmente al corrente del fatto che con la versione 1.0 i programmatori avevano deciso d'includere le routine software per convertire le immagini in scala di grigi in immagini IFF a 16 colori. Come ho già detto nel mio articolo del numero precedente, questa caratteristica sarà un'opzione accessoria dal costo addizionale.

Le routine di conversione possono

creare immagini IFF in bassa risoluzione (320 per 400 pixel) partendo dai sedici possibili grigi dell'Amiga. Le routine possono anche creare immagini in alta risoluzione (640 x 400 pixel) con 31 grigi utilizzando il dithering dei 16 colori base. Dal momento che lo scanner genera le sue immagini dithered con una matrice di 6 x 6 punti, le immagini IFF risultanti dalla conversione hanno una larghezza di circa soli 260 pixel sul display in bassa risoluzione e di 520 su quello in alta. Quando si fa la conversione in scala di grigi, si ha la possibilità di conservare il rapporto originale di scansione del pixel quadrato o di modificare le proporzioni dell'immagine sulla base dello schermo NTSC. Le proporzioni non modificate funzionano bene con il display PAL e vanno benissimo per applicazioni che richiedono come risultato finale l'hard copy su carta. Dopo aver giocherellato un po' con le regolazioni dello scanner, sono stato in grado di creare alcune immagini sullo schermo dal risultato davvero superbo.

Il mese scorso ho finalmente ricevuto la mia copia di *Professional Draw 2.0* della Gold Disk. *Pro Draw 2.0* include la possibilità di far scorrere il testo accanto a una linea curva (testo sagomato) e quella d'importare immagini Postscript (EPS). Le immagini EPS appaiono come semplici box che possono essere dimensionati a piacere, in ogni caso le immagini non possono essere visualizzate sullo schermo. Nel package è inclusa anche un'utility di autotracciamento, che converte la grafica bit-map in oggetti strutturati. Le mie prime impressioni sono che *Professional Draw* è un potente programma per il disegno strutturale che ha bisogno di un po' di tempo da parte del suo utente per poter essere utilizzato a fondo. Buona parte delle difficoltà di utilizzo deriva dall'eccessiva brevità del manuale. Per esempio, la descrizione di come far allineare il testo a una curva non dice che per far funzionare l'opzione si devono selezionare sia la stringa di testo sia la curva. Anche se dopo aver usato per un po' il programma la procedura risulta ovvia, nel principiante la scarnezza del manuale può suscitare confusione.

Il manuale dice anche che il testo segue le curve disegnate con il tool "pen". Ho verificato che sono in grado di guidare il testo anche curve chiuse, come quelle create con il tool "ellipse". I punti d'inizio e di fine della curva determinano la direzione nella quale il testo scorre. La base è sempre tracciata in direzione della curva. Se si desidera collocare la parte superiore del testo lungo la curva, si deve far scorrere il testo lungo una seconda curva che è concentrica rispetto alla prima. Una volta che ho capito come districarmi tra questi intrecci, la manipolazione del testo e delle curve si è rivelata veloce e di semplice gestione.

Il 1990 dev'essere l'anno della versione *3.qualcosa* per quasi tutto quello che è disponibile per l'Amiga per più di due anni (naturalmente a eccezione dell'AmigaDOS). La SunRize Industries ha realizzato di recente la versione 3.1 del suo campionatore Perfect Sound e del relativo software, mentre la versione 3.11 dovrebbe essere disponibile nel momento in cui leggete questo articolo. L'hardware del campionatore è ora inserito all'interno di un package in plastica, al posto del quasi indistruttibile contenitore metallico originale. Oltre al nuovo indirizzo della SunRize Industries e alla sistemazione di alcuni bug, la versione 3.1 include funzioni di eco e delay in tempo reale sui suoni campionati. I cambiamenti principali della versione 3.11 riguardano invece la compatibilità con le schede 68020 e 68030 e l'Amiga 3000.

È passato ormai un po' di tempo da quando ho ricevuto la versione 2.0 di *Excellence!* e la 3.0 di *ProWrite* e ho lavorato con questi programmi a sufficienza per parlarvi di alcune importanti conclusioni. Lungo il cammino ho trovato alcuni bug sul modo in cui *Excellence!* miscela grafica e testo; ebbene, non era passata neanche una settimana da quando avevo informato la Micro Systems Software che già avevo ricevuto una nuova versione del programma con gli errori corretti. Questo vuol dire che alla MSS forniscono un prezioso supporto-utente.

L'inclusione di grafica in un documento realizzato con *Excellence!* è un po' limitata. Anche se *Excellence!* è in grado d'importare quasi qualsiasi immagine IFF, poi la tratta come se fosse un singolo carattere di testo. Questo vuol dire che la spaziatura, per la linea che contiene la grafica, diventa la stessa dell'altezza della grafica. Il risultato è che bisogna accettare o uno scomodo layout di pagina o restringere le dimensioni e il posizionamento della grafica nel testo.

D'altro canto, *ProWrite* gestisce bene la grafica IFF. Offre sei modi di convertire sia alla palette di *ProWrite* sia in bianco e nero o grigio. Le opzioni per la conversione in grigio includono due livelli di dithering. La grafica importata si sovrappone semplicemente al testo con il colore di sfondo trattato come trasparente. Tutto quello che è richiesto per completare il layout è la regolazione dei margini, l'aggiunta di spazi ad alcune linee, lo scorrimento di alcuni paragrafi e il posizionamento della grafica come richiesto. Anche se il testo non può essere fatto scorrere automaticamente intorno alla grafica, come si può fare con un completo programma d'impaginazione, l'interfaccia WYSIWYG di *ProWrite* rende facile ottenere esattamente il layout desiderato.

Gli stampati di testo e grafica di *ProWrite* si basano su una risoluzione orizzontale di 80 punti per pollice per 72 di risoluzione verticale. Questa risoluzione, o alcuni suoi multipli, è disponibile su molte stampanti a matrice di punti a 8 aghi, mentre invece diverse 24 aghi non supportano questi valori. *ProWrite* cerca di regolare le proporzioni dello stampato sulla base della risoluzione della stampante posseduta, cosa che può distorcere le fonti bit map utilizzate per gli stampati a matrice di punti. Il programma consente anche di disattivare la regolazione delle proporzioni mantenendo nel contempo una corretta visualizzazione WYSIWYG. Il risultato finale di tutto questo è che se la vostra stampante non supporta la risoluzione specificata, si può rivelare necessario dover sperimentare un po' per ottenere i risultati desiderati.

Se comparato con *Excellence!*, *ProWrite* offre anche una maggiore versatilità nell'uso delle fonti near letter quality (NLQ) interne alle stampanti. Per esempio, con la stampante Panasonic KX-1124 a 24 aghi, che usa il driver EpsonQ, il programma supporta le versioni normale e allargata nelle densità di stampa Pica, Elite e Ultra Condensed. Sono incluse le fonti bit map che corrispondono a queste densità di stampa. Dal momento che *ProWrite* utilizza i driver di stampa dell'Amiga, le densità di stampa NLQ variano da stampante a stampante.

Per molti altri versi, *Excellence!* e *ProWrite* sono comparabili. Entrambi i programmi dispongono di uno spell checker (per la lingua inglese). *Excellence!* dispone anche di un utile

controllo grammaticale e delle frasi. Personalmente, ritengo difficile effettuare una scelta tra questi due programmi. Se la gestione della grafica è un'esigenza primaria, *ProWrite* è chiaramente il vincitore. In ogni caso, mi piace molto anche il controllo ortografico di *Excellence!* (io però sono americano... e a voi lettori italiani può interessare poco un controllo ortografico dell'inglese). Bisogna anche tenere presente che il prezzo di *Excellence!* è superiore, ma che questo programma dispone di un output PostScript come caratteristica standard; *ProWrite* richiede invece *ProScript*, un'utility separata della New Horizons, che ovviamente ha un suo costo addizionale e che non è in grado di gestire documenti con più colonne sulla stesssa pagina.

La versione più recente di *ProWrite* è la 3.1. Le sue nuove caratteristiche sono descritte in un addendum di sette pagine. Il requester dei file è stato così migliorato che ora è in grado di elencare tutti i dischi installati e i vari assegnamenti dei percorsi. È stato anche aggiunto un selettore che consente di attivare la visualizzazione di tutti i file, dal momento che normalmente il programma visualizza solo i file *ProWrite*. La versione 3.1 include anche un convertitore del testo in voce sintetizzata dell'Amiga, utilizzando così le capacità di sintesi vocale della macchina. È possibile inoltre scegliere tra una voce maschile, femminile e robotica, con controllo su velocità e tonalità. Se si dispone dell'AmigaDOS 2.0, dell'ECS e di un monitor VGA o multisync, *ProWrite* consente ora di utilizzare un display non-interlace di 640 x 480 pixel o 1280 x 400.

Software per gioco e profitto

Ho avuto di recente l'opportunità di giocare un po' con *Art Department* della ASDG. Si tratta di un programma di image processing potente e facile da usare basato sul software sviluppato dalla ASDG per lo scanner a colori Sharp JX-100. Questo image processor utilizza 24 bit per pixel per le immagini a colori (16.777.216 colori) e 8 bit per pixel per le immagini a scala di grigi (256 livelli). Le immagini possono essere convertite e ridimensionate per tutti i modi di visualizzazione dell'Amiga, inclusi i formati NTSC e PAL. La versione basica di *Art Department* (89,95 dollari) è dotata di un modulo per il caricamento di immagini IFF con fino a 24 bit plane e per immagini a 21 bit di DigiView. Il modulo di caricamento IFF supporta immagini in tutti i formati IFF, incluso EHB, HAM, SHAM, dymanic HAM, ARES e dynamic HIRES.

A un costo addizionale, sono disponibili moduli di caricamento per altri formati d'immagini. I moduli disponibili includono Sculpt e Turbo Silver a 24 bit, Rendition a 32 bit, GIF a 8 bit, Targa a 24 bit (formati 1, 2, 9 e 10) e i file MS-DOS a 256 colori di *Deluxe Paint Enhanced*. I prezzi di questi moduli vanno dai 19,95 ai 39,95 dollari. Nel momento in cui leggete questo articolo sono anche disponibili moduli per i file d'immagini TIFF e PCX. La ASDG sta insomma facendo tutti gli sforzi possibili per offrire moduli di caricamento per tutti i formati di file d'immagini conosciuti disponibili sui vari computer.

Durante il mio giocherellare su *Art Department*, ho iniziato con una collezione d'immagini GIF a 256 colori che ho preso da una banca dati MS-DOS. Ho convertito queste immagini in immagini Amiga IFF e poi ho usato la scala di grigi. Le ho trasformate poi in immagini a due colori, in bianco e nero, in alta risoluzione, in interlace, dithered rimanendo sempre fedele alle tonalità originali. Ho poi generato un'immagine a 16 colori in alta risoluzione interlace e utilizzando le matrici dither ho potuto mantenermi estremamente vicino alla colorazione originale. *Art Department* mi ha dato ottimi risultati senza che fosse necessario intervenire sui controlli di luminosità, contrasto e bilanciamento colore pur disponibili. Il programma è anche in grado di separare i colori delle immagini IFF.

La più recente proposta della ASDG è *Art Department Professional* (199,95 dollari). Perry Kivolowitz della ASDG ha dimostrato questo programma al meeting AMUSE tenutosi a metà agosto. Questa versione supporta l'Arexx, importa e salva sia in formati Amiga che non-Amiga (la versione originale salva solo nei formati Amiga) e fornisce controllo di scanner e altro hardware grafico. Perry ha addirittura affermato che la ASDG realizzerà il software di controllo e la relativa interfaccia hardware per qualsiasi apparecchio per la gestione delle immagini l'utente desideri. Tutto questo in 30 giorni e fornendo in prestito l'unità alla ASDG!

Oltre a una versione preliminare di *Art Department Professional*, Perry ha fatto vedere una collezione d'immagini, su diapositive 35 mm, create con questo programma con una Polaroid palette CI-3000 collegata. La CI-3000 è in grado di generare immagini con una risoluzione che arriva sino a 2.000 per 1.500 pixel. Le diapositive ritraevano almeno quattro immagini HAM realizzate con *Art Department Professional* su una singola super bit-map per l'uso con la CI-3000. Queste immagini hanno fatto finire con rapidità le mie calze in lavanderia....

Art Department Professional è provvisto di moduli per il caricamento e il salvataggio nei formati DigiView a 21 bit, Sculpt, Silver, GIF, PCX, DeluxePaint II Enhanced e MacPaint. Il caricatore Super-IFF, oltre a tutti i formati standard IFF, supporta anche EHB, HAM, AHAM, SHAM, dynamic HAM, ARES e dynamic HIRES. Il software separato *Art Department Professional Conversion Pack* (89,95 dollari) include moduli per i formati Targa, TIFF Pict2 e Rendition.

Chi già possiede la versione standard può passare a quella Professional per 75 dollari. Se la vostra attività include la realizzazione d'immagini a colori di alta qualità, allora *Art Department Professional* deve entrare nella vostra prossima lista della spesa.

Materiale hardware

Ai primi di settembre la Great Valley Products (GVP) ha iniziato a promuovere la sua nuova linea di schede serie II. Sulla base dei nuovi circuiti a integrazione su larga scala sviluppati dalla Motorola, la serie II mostra uno stile progettuale che appare impeccabile. Il nuovo chip ha ridotto la necessità di componenenti al punto che la serie II offre un controller con RAM fino a 8 MB (usando chip da 1 megabit per 8 SIMM) e un hard disk, il tutto su un'unica scheda per A2000 senza che quest'ultima appaia assolutamente affollata. La GVP ha inoltre presentato il suo A-Net, un controller di rete locale Ethernet per l'Amiga.

Per non essere da meno, la Supra ha recentemente annunciato il suo nuovo hard disk + RAM SupraDrive 500XP. Una nota positiva su questa periferica è che è stata disegnata in

modo da non aver bisogno di un trasformatore esterno. Nel 500XP possono essere installati fino a 2 MB di RAM con chip da 1 megabit oppure 8 MB con chip da 4 megabit. I prezzi partono da 679 dollari per l'hard disk da 20 MB e 512K di RAM.

Sul versante della "mela Amiga", posso dirvi che A-Max II ora include il supporto per hard disk GVP, IVS TrumpCard e Commodore A2090 e A2091/A590. Il supporto per l'hard disk è gestito attraverso un driver contenuto nella directory devs. L'hard disk viene formattato con una partizione riservata ad A-Max. Quando funziona sotto A-Max, il Macintosh riformatta poi questa partizione. Anche se il numero di hard disk supportati è ancora basso, quando saranno disponibili, si potranno aggiungere nuovi driver semplicemente includendoli nella directory devs. A-Max è stato testato con versioni del sistema Macintosh fino alla 6.0.5. *Dulcis in fundo*, A-Max II gestisce la maggior parte dei suoni digitalizzati del Macintosh. Gli unici problemi possono arrivare con quel software che pilota direttamente l'hardware sonoro del Macintosh.

È veramente necessario?

La software Support International, che si è fatta conoscere per i suoi copiatori per C-64, ha annunciato un prodotto di questo tipo per l'Amiga. *Maverick* per Amiga (39,95 dollari) dispone di oltre 200 parametri di copia e include un copiatore di dati ad alta velocità e un copiatore a parametri. *Maverick* dovrebbe essere in grado di rimuovere le protezioni anti-copia al livello di permettere la registrazione su hard disk dei programmi protetti.

Giochi

Sul vecchio e glorioso Commodore 64, *Pinball Construction Set* era uno dei miei programmi preferiti. Quando l'Amiga venne introdotto sul mercato, la Electronic Arts dichiarò che avrebbe realizzato una versione di *Pinball Construction Set* per l'Amiga. Fotografie di schermate di diverse versioni preliminari vennero anche riprodotte da più di una rivista. Sfortunatamente però, la Electronic Arts non arrivò mai a pubblicare davvero la versione a 16 bit di *Pinball Construction Set*.

Dopo cinque anni di attesa, è finalmente pronto un degno sostituto. Giudicando dai suoi giochi di esempio, *Power Pinball* della Karmasoft sembra proprio quello per cui ho aspettato. Per gli sfondi il gioco usa immagini IFF a 32 colori in bassa risoluzione. Gli effetti sonori e la musica di sottofondo si basano su file contenenti campioni IFF 8SVX che si possono creare con qualsiasi digitalizzatore per l'Amiga. Con *Power Pinball* si possono personalizzare gli schermi di gioco di esempio, crearne di completamente nuovi o semplicemente giocare con quelli inclusi nel programma. Sfortunatamente, *Power Pinball* non crea giochi stand-alone distribuibili liberamente; per giocare è sempre necessaria una copia del disco programma. Se siete amanti del flipper amerete *Power Pinball*.

I simulatori di combattimento aereo sembrano essere diventati un nuovo genere molto prolifico. Ho appena ricevuto una copia di *Wings* della Cinemaware. L'ambientazione è quella della Prima guerra mondiale nella quale il giocatore è il pilota di un biplano dell'epoca. Dal momento che i controlli di volo non consistono in nient'altro che nel joystick, mi sono trovato a mio agio dopo solo pochi minuti dal caricamento. Come per tutte le produzioni Cinemaware, le sequenze di apertura di *Wings* sono superbe anche se stridono con la grafica un po' abbozzata delle reali sequenze di volo. D'altro canto, i dettagli dell'aeroplano e del suo pilota sono eccellenti.

Durante il gioco, il punto di vista è solitamente da dietro il pilota che guarda sempre in direzione dell'aeroplano nemico. Il punto di vista passa all'aeroporto durante le collisioni o altri disastri. Sono disponibili oltre 230 missioni con le quali formare il proprio personaggio: pattugliamenti, scorte, bombardamenti... Alcuni messaggi dal comandante aiutano a mantenere una sensazione di realismo. Sono anche tenute in memoria delle statistiche sulle prestazioni del proprio personaggio e della squadriglia. La confezione comprende un manuale d'istruzioni di quattordici pagine e un manuale illustrato per il pilota di settantasei pagine che è ricco di dettagli sulla Prima guerra mondiale. Ho trovato *Wings* interessante, informativo e divertente da giocare. Il biplano della Prima guerra mondiale è ben sintonizzato sulla mia velocità.

Arrivi dell'ultimo minuto

Ho finalmente ricevuto la mia copia di *PageStream 2.0* e non ho avuto neanche il tempo sufficiente per inserire le 60 nuove pagine nel manuale. Maggiori dettagli su *PageStream 2.0* nel prossimo numero.

World of Commodore

A questo punto, avrei dovuto raccontarvi qualcosa su una manifestazione che si è tenuta i giorni 15 e 16 di settembre. Pensate: ho dovuto scegliere tra il Commodore Show e la "Book Country street fair" di New York. Sarei dovuto andare alla fiera del libro. Infatti, dopo aver guidato per 110 miglia fino al Centro Congresi di Valley Forge, ho scoperto che la fiera era stata degradata a una sagra di club di utenti Commodore e che gli organizzatori (l'Hunter Group) non si erano neanche degnati di stampare un elenco degli espositori. La maggior parte degli stand erano occupati da club con una spruzzatina di programmatori di C-64/128 qua e là. Nuovi prodotti per l'Amiga erano virtualmente inesistenti.

Lo stand più interessante era quello occupato dalla Pulsar di Dr. Oxide, che ha passato più di 15 minuti lamentandosi della spedizione di uno scanner manuale a 4096 colori che la Federal Express era riuscita a perdere in qualche punto tra gli Stati Uniti e la Germania. Dr. Oxide aveva comunque in fiera un prototipo funzionante del suo scanner manuale a colori da 995 dollari. Lo scanner a colori ha una scansione di 64 mm d'ampiezza a 90 punti per pollice, il che si traduce in circa 226 punti, meno dei 320 punti di larghezza dello schermo in bassa risoluzione dell'Amiga. Con una singola passata è in grado di scansire 4096 colori. L'interfacciamento con il computer è effettuato attraverso la porta di espansione dell'Amiga 500. È anche disponibile una versione su scheda per l'Amiga 2000. Lo scanner è venduto unitamente al suo software per la scansione e a *Deluxe Photolab* della Electronic Arts. Il prodotto sarà disponibile nel momento in cui leggerete questo articolo.

Alla fiera ha fatto la sua comparsa anche la rivista *Compute* di ottobre. 264 pagine con una sezione MS-DOS, una Amiga, una C-64 e una Macintosh. Una cinquantina di pagi-

ne erano rispettivamente dedicate a MS-DOS e Amiga, quaranta al C-64 e solo otto per il Mac. Chi era abbonato a una delle riviste che ora formano le varie parti di *Compute* riceverà solo la parte pricipale e la sottosezione (che prima era una rivista autonoma).

E infine un annuncio: io rappresento i vostri occhi e le vostre orecchie sul mondo statunitense. Se avete delle richiste da farmi su quello che devo trattare nei prossimi numeri scrivete pure a *Commodore Gazette*. Nel frattempo io sono già al lavoro. A presto. ■

Per ulteriori informazioni contattare direttamente:

ASDG Incorporated
925 Stewart Street
Madison, WI 53713
USA
(Tel. 001/608/2736585)

Cinemaware Corp.
4165 Thousand Oaks Blvd.
Westlake Village, CA 91362
USA
(Tel. 001/805/4956515)

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
USA
(Tel. 001/415/5717171)

Gold Disk
5155 Spectrum Way, Unit 5
Mississauga, Ontario
Canada L4W 5A1
(Tel. 001/416/6024000)
(Fax 001/416/6024001)

Great Valley Products
600 Clark Ave.
King of Prussia, PA 19406, USA
(Tel. 001/215/3378770)

KarmaSoft
P.O. Box 1034
Golden, CO 80402, USA
(Tel. 001/303/2771241)

Micro Systems Software
12798 Forest Hill Blvd.
Suite 202
West Palm Beach, FL 33414, USA
(Tel. 001/407/7900770)
(Fax 001/407/7901341)

Migraph, Inc.
200 S. 333rd, Suite 220
Federal Way, WA 98003
USA
(Tel. 001/206/8384677)

New Horizons Software, Inc.
P.O. Box 43167
Austin, TX 78745
USA
(Tel. 001/512/3286650)
(Fax 001/512/3281925)

NewTek
215 East 8th St.
Topeka, KS 66603
USA
(Tel. 001/913/3541146)

Pulsar
410 Maple Ave.
Westbury, NY 11590, USA
(Tel. 001/516/9976903)

Software Support International
2700 NE Andresen Road
Suite A-10
Vancouver, WA 98661
USA
(Tel. 001/206/6951393)

SunRize Industries
270 E. Main St. Ste C
Los Gatos, CA 95030
USA
(Tel. 001/408/3543488)
(Fax 001/408/3542596)

Supra Corp.
1133 Commercial Way
Albany, OR 97321
USA
(Tel. 001/503/9679075)

COMINCIAMO DA ZERO: PARTE QUARTA

Il Workbench non rappresenta l'unica possibilità di lavorare con l'Amiga. Lo imparerete presto grazie allo Shell

di Mark L. Van Name e Bill Catchings

I precedenti articoli di questa serie sono stati pubblicati sui numeri 2-3-4/90 di Commodore Gazette.

Nei primi tre articoli di questa serie, ci siamo concentrati esclusivamente sul *Workbench*, l'interfaccia-utente grafica dell'Amiga. Il computer ha però anche una seconda interfaccia standard, denominata Shell. Anche se l'approccio con lo Shell può sembrare all'inizio meno intuitivo e più difficoltoso, scoprirete poi che lo Shell permette di fare alcune cose non possibili con il *Workbench*. Che aspettiamo allora? Lanciamoci ad apprendere un nuovo modo di lavorare con il nostro Amiga.

Lo Shell vi parla da qui

Per far apparire lo Shell, bisogna aprire il disco del *Workbench* (un doppio click sull'icona del disco è il modo più semplice). Lo Shell è semplicemente uno strumento su quel disco. Nella parte sinistra della finestra standard del *Workbench 1.3*, vedrete la sua icona rettangolare denominata "Shell" con i caratteri "1>" disegnati nell'angolo superiore sinistro. Aprite questo strumento come qualsiasi altro (di nuovo, un doppio click è il modo più semplice). Apparirà così una finestra a piena larghezza delle dimensioni verticali di quasi un terzo dello schermo. Siete ora nello Shell, pronti per lavorare.

Quando vi trovate nello Shell dovete abituarvi a un ambiente non grafico. La finestra dello Shell non contiene icone; infatti è vuota, eccetto per la richiesta comandi "1>" nell'angolo superiore sinistro. A differenza del *Workbench*, lo Shell è un'interfaccia linea comando (CLI). Nelle precedenti versioni dell'AmigaDOS, il CLI era l'unica interfaccia alternativa: si trattava infatti di una versione meno potente dello Shell che si può ancora trovare nel cassetto System della finestra del *Workbench*. Dal momento che è migliore del CLI, vi invitiamo comunque ad affezionarvi allo Shell.

Un'interfaccia linea comando è un interprete di comandi che devono essere impartiti dall'utente. Si impartiscono digitandoli e premendo il tasto Return per dire allo Shell d'iniziare a lavorarci sopra. Il carattere "1>" viene detto richiesta comandi perché indica che lo Shell è pronto per un comando. La parte "1" della richiesta è il numero di questo Shell in relazione ad altre finestre Shell eventualmente attive. Bisogna infatti ricordarsi che l'Amiga è un sistema multitasking e che ci possono quindi essere più Shell attivi nello stesso tempo. Se è così, il secondo Shell ha una richiesta "2>", il terzo "3>" e così via. Ma diamo il nostro primo comando. Digitate:

DIR

e premete Return (*nota:* da questo punto in avanti assumeremo che voi sappiate che dovete premere Return dopo un comando. Inoltre, nonostante i comandi siano qui indicati in lettere maiuscole, voi potete inserirli anche in lettere minuscole; non fa nessuna differenza quando si usa lo Shell). Apparirà una lista di parole. DIR, che è l'abbreviazione di Directory, vi mostra i nomi dei file sul disco.

Avrete notato che l'elenco di file non ha trovato sufficiente spazio nella finestra, e che lo Shell lo ha fatto scorrere automaticamente verso l'alto. Più avanti, quando avrete maggiore confidenza con lo Shell, una finestra piccola non vi creerà problemi. Adesso però, per i nostri fini didattici, è meglio che abbiate la possibilità di vedere visualizzata la maggiore quantità d'informazioni possibile; quindi procedete allungando la finestra dello Shell a tutto schermo (fate come con qualsiasi altra finestra: usate la barra nella parte superiore per portare la finestra alla sommità dello schermo, e portare il gadget di dimensionamento nella parte inferiore destra della finestra fino alla parte inferiore destra dello schermo. Sia per la barra, sia per il gadget, l'operazione si esegue puntandoci sopra con la freccetta del mouse e premendo e tenendo premuto il tasto destro del mouse durante il movimento).

I segreti dello Shell

Notate che la richiesta 1> appare nuovamente sulla linea successiva dopo l'elenco dei vostri file su disco. Questo vuol dire che lo Shell è pronto per un altro comando. Digitate:

DIR

in modo da poter vedere di nuovo l'elenco di file. Alcuni dei nomi di questa lista dovrebbero esservi vagamente familiari, dal momento che sono gli stessi nomi dei cassetti, strumenti e progetti che avete già visto nella finestra del *Workbench 1.3*.

Lo Shell però, non li chiama cassetti, strumenti e progetti. Ha infatti dei suoi nomi quasi per tutto, anche se usa la stessa organizzazione di base del *Workbench*. Digitate:

INFO

I dati risultanti vi danno alcune utili informazioni sui dischi che sono correntemente nel vostro Amiga. Per rendere le cose semplici, esamineremo solo cosa viene mostrato sul disco del *Workbench 1.3* che assumeremo essere nel disk drive interno.

Da quanto è visualizzato sullo schermo si deduce che lo Shell ha due nomi per ogni disco: uno sotto la voce "Unit" e un altro sotto la voce "Name". Trovate la riga con "DF0:" sotto Unit. DF0: è il nome dell'unità che rappresenta il vostro disk drive interno. Potete usare il nome di unità DF0: ogni volta che volete riferirvi al disco presente nel disk drive interno. In modo analogo, se avete un disk drive esterno potete fare riferimento al disco inseritovi con DF1:. Potete anche riferirvi ai dischi tramite il nome che appare sotto la voce "Name". In questo momento, dallo Shell potete chiamare il vostro disco *Workbench* sia "DF0:" sia "Workbench 1.3:".

Notate i due punti (:) dopo il nome *Workbench* nella riga precedente. Come i due punti alla fine del nome di unità DF0:, devono essere inclusi perché dicono allo Shell chi vi state riferendo a un disco.

Dal punto di vista dello Shell, l'interno di ogni disco è una collezione di zero o più file e directory. Le directory sono la versione dello Shell dei cassetti; una directory può contenere file e altre directory. Potete riconoscere le directory nell'elenco che avete ottenuto con DIR dall'estensione "(dir)" che appare alla destra di ognuna di esse. Lo stesso disco agisce come una sorta di directory "radice"; tutto quello che non è in una directory con un nome specifico si trova nella directory radice.

Tutto ciò che non è un disco o una directory è un file, a prescindere che il file, in accordo con la terminologia del *Workbench*, sia un progetto o uno strumento. Anche le icone sono file. Se guardate attentamente l'elenco della directory, noterete che per ogni directory che appare nella finestra del *Workbench* – per esempio, la directory System – c'è un file con quel nome più un suffisso ".info". System ha System.info, Utilities ha Utilities.info, e così via. Per convenzione, tutti i programmi dell'Amiga chiamano i file delle icone in questo modo.

È buona norma non mettere in fondo al nome di un file l'estensione ".info", a meno che non sia un'icona. Se volete potete anche farlo – lo Shell non ve lo impedirà – ma questo potrebbe confondere chiunque altro abbia bisogno di guardare o anche lavorare con i vostri file. Inoltre, prestate molta attenzione prima di cancellare qualsiasi file .info, dal momento che un file senza un file .info per l'icona non è visibile dal *Workbench*. Per esempio, se cancellate il file Utilities.info, non sarete più in grado di vedere il cassetto Utilities dal *Workbench*. Naturalmente, ci sono file che potete vedere *solo* dallo Shell. Notate, per esempio, che il vostro elenco di file visualizzato con DIR, mostra delle directory chiamate "c" e "s" che non potete vedere dal *Workbench* perché non hanno icone e quindi nessun file .info.

Nelle parti precedenti di questa serie "Cominciamo da zero", abbiamo visto come svolgere alcune operazioni di base da *Workbench* su dischi, cassetti, strumenti e progetti. In modo simile, ci sono alcuni comandi basici dello Shell, che vi permettono di lavorare su dischi, directory e file. La prossima volta, parleremo di alcuni dei più importanti di questi comandi, a anche di alcuni trucchetti per impartirli.

PROGRAMMARE IN C

I PROCESSI E LE STRUTTURE DELL'AMIGADOS

Che cos'è un processo? Quali caratteristiche fanno di un codice un processo? Le risposte a questi quesiti, la discussione delle quattro più importanti strutture del DOS, un utile programma: tutto questo nelle prossime righe

di Eugene P. Mortimore

In alcuni articoli precedenti abbiamo descritto il file system dell'AmigaDOS e abbiamo illustrato come usare le funzioni DOS per manipolare i file nei programmi in C. Non abbiamo però descritto dettagliatamente il funzionamento interno del DOS e i meccanismi software che vengono attivati durante una chiamata alle sue funzioni.

L'importanza di questi meccanismi di comunicazione da programma a programma e da programma a sistema, e altri importanti concetti correlati, ci portano a un'ampia descrizione dei processi dell'Amiga. La comprensione dei processi è essenziale per poter capire il funzionamento interno, e generalmente nascosto, del complesso sistema di scambio delle informazioni che avviene di continuo nel sistema dell'Amiga.

In questo articolo spiegheremo innanzitutto i concetti che definiscono i processi dell'Amiga e successivamente forniremo ampie e precise definizioni dei vari parametri presenti nelle quattro strutture più importanti del DOS, quelle cioè che hanno un ruolo basilare nelle procedure di gestione dei processi stessi. Noterete che queste procedure usano dei meccanismi a liste concatenate e assomigliano molto alle procedure di gestione della struttura ExecBase che abbiamo trattato in un articolo precedente. Infine, forniremo un interessante e utile programma che potrete usare per determinare quali dispositivi, e processi associati, sono correntemente attivi nel sistema e come poter accedere ai parametri che li definiscono.

Definizione di processo

Che cos'è un processo? E quali caratteristiche fisiche e logiche fanno di un particolare codice eseguibile, e di un gruppo di strutture, un processo? Rispondere a questi quesiti è importante in quanto vi aiuta a capire come i processi coesistono nel sistema, come nascono, come comunicano fra di loro per scambiarsi dati e informazioni e perché le strutture del DOS sono definite in un certo modo.

Osservando il diagramma della Figura 1 notiamo un rettangolo che rappresenta un processo (routine software del processo) con un input (ingresso) sulla sinistra e un output (uscita) sulla destra. In altre parole il rettangolo rappresenta una serie di generiche attività software (l'esecuzione di una serie di routine interne al processo)

Figura 1: Caratteristiche generali di un processo dell'Amiga

che prendono delle informazioni da un input per produrre delle informazioni in output. In generale, possiamo distinguere, così come vengono distinti nei file di inclusione del DOS, tre tipi di processi e relativi dispositivi: reali (o fisici), logici e di volume. La definizione e le caratteristiche che distinguono questi tre tipi di processi risulterà più chiara durante la lettura di questo articolo.

I processi dei dispositi console e seriale

Esaminiamo per esempio un processo del dispositivo Console (CON:, RAW: oppure NEWCON:). In questo caso l'input del processo proviene dall'utente tramite la tastiera e l'output viene visualizzato sullo schermo. Il processo consiste in tutte le routine interne del dispositivo console che leggono i segnali elettrici di scansione della tastiera e li convertono in caratteri visualizzabili sullo schermo. Agli utenti tutto ciò sembra ovvio e trasparente ed è facile dare per scontate le routine

interne al processo.

Il caso specifico del processo associato al dispositivo seriale (SER:) offre un altro buon esempio di processo. In questo caso i dati provengono o vengono inviati tramite un dispositivo seriale fisico, la porta seriale, e vengono collocati (o prelevati, a seconda del tipo di operazione) in appropriati buffer in memoria. In altre parole, durante una ricezione di dati dalla porta seriale l'input del processo corrisponde ai dati provenienti dal dispositivo fisico esterno collegato alla porta, e l'output corrisponde ai dati che vengono collocati nel buffer, mentre durante una trasmissione avviene l'opposto.

È importante notare che questi dati possono sottostare a specifiche trasformazioni imposte dall'intervento delle routine interne del dispositivo seriale. Tale considerazione può essere fatta anche a riguardo dei processi del dispositivo parallelo dell'AmigaDOS (PAR:).

Prendiamo in esame per esempio il processo del dispositivo associato alla stampante (PRT:). In questo caso l'input è generalmente il contenuto di un buffer definito da un programma, cioè i dati che l'utente vuole stampare, e l'output è il testo scritto sulla carta dalla stampante. Alcuni caratteri contenuti nel buffer possono però essere caratteri di controllo della stampante ed è compito delle routine interne al processo del dispositivo PAR: interpretare ed elaborare in maniera corretta questi caratteri distinguendoli da quelli normalmente stampabili.

Scambio d'informazioni tra processi

Nella Figura 2 viene schematizzata la comunicazione tra due processi. Quello sulla sinistra potrebbe essere un processo creato da un vostro programma, con il proprio codice eseguibile, e quello sulla destra potrebbe essere uno dei processi predefiniti dal sistema, per esempio un processo di gestione file del dispositivo DF0: o DF1:. In questo caso l'obiettivo del processo del vostro programma è di richiedere informazioni all'altro processo per continuare in seguito la propria esecuzione.

Per prima cosa notiamo che entrambi i processi della Figura 2 possiedono delle message port. Se il vostro programma ha creato un processo, la sua message port è rappresentata dal parametro pr_MsgPort nella struttura Process relativa a quel processo. Il vostro programma dovrebbe quindi avere allocato e inizializzato esplicitamente una struttura Process per ogni specifico processo creato.

La message port del processo di gestione file (file handler), per esempio quello relativo a DF0:, viene definita internamente quando le routine del sistema creano automaticamente tale processo, durante il boot, oppure viene creata usando le specifiche contenute nel file Mountlist, presente nella directory DEVS:, mediante il relativo comando Mount. Ciascuna di queste message port, indifferentemente dal loro uso o dai processi a esse associati, usano un segnale Exec, e più precisamente il signal bit numero otto, per informare il processo all'arrivo di un messaggio. Questa è una convenzione standard di tutti i processi dell'Amiga. Quindi se il vostro programma invia un messaggio al processo di sistema associato a DF0:, per esempio per aprire un file chiamando la funzione Open della DOS Library, le routine interne del DOS invieranno un messaggio a tale processo, chiamato file handler, e l'arrivo di questo messaggio attiverà il signal bit numero otto, causando una serie di azioni software predisposte per l'apertura del file. Lo stesso tipo di passaggio avviene quando il vostro programma chiede al processo DF0: di leggere dei dati da un file aperto, chiamando la funzione Read del DOS, o di chiudere un file tramite la funzione Close.

Riassumendo, ogni volta che il vostro programma chiama una funzione DOS, il DOS automaticamente inoltra la richiesta, sotto forma di messaggio (DOS message packet), alla message port dell'appropriato processo file handler causando l'attivazione del signal bit numero otto. Il processo file handler, dopo aver elaborato la richiesta, rispedisce un messaggio alla message port del programma e ne attiva il signal bit numero otto. Il messaggio di ritorno riporta le informazioni richieste che permettono al programma di continuare l'esecuzione.

Un altro modo di richiedere informazioni a un processo file handler, oltre a quello appena descritto, è quello di usare direttamente i message packet del DOS. È importante capire che, indifferentemente dal metodo usato dal vostro programma, il sottostante sistema software del DOS converte tutte le richieste del vostro programma in richieste di tipo message packet e tutte le comunicazioni avvengono tramite le message port usando il signal bit numero otto per indicarne l'arrivo.

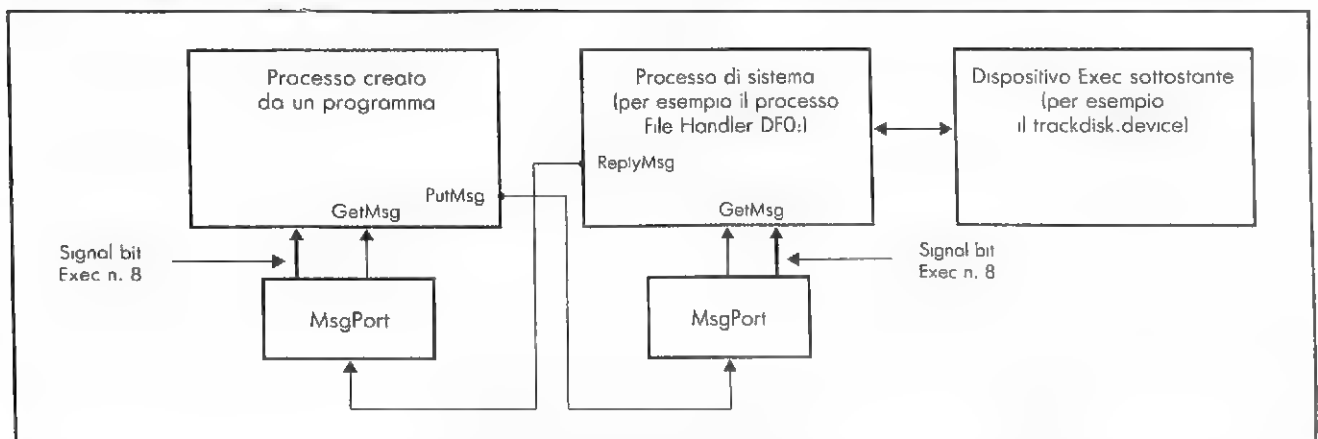



Figura 2: Le relazioni che intercorrono tra un processo di programma, un processo di sistema e un dispositivo dell'Exec

Per comprendere meglio questo concetto ricordiamo che la funzione Open del DOS restituisce un puntatore a una struttura FileHandle, e che tale struttura, definita nel file include dosextens.h, contiene sia le informazioni delle strutture Message e MsgPort, sia quelle relative al DOS packet.

Comunicazione tra un processo e i dispositivi Exec

Nella Figura 2 notiamo anche che un processo DOS può comunicare con i sottostanti dispositivi Exec. Per esempio, tutti i processi file handler (DF0:, DF1:...) comunicano con le routine di gestione del trackdisk.device; il dispositivo SER: del DOS comunica con il serial.device dell'Exec; il dispositivo PAR: comunica con il parallel.device; i dispositivi CON: e RAW: comunicano con il console.device. Per il momento non importa conoscere i dettagli di questa comunicazione. Ci basti sapere che esiste, e che ciò rende più facile la costruzione di altri dispositivi DOS poiché i sottostanti, e sempre presenti, dispositivi Exec eseguono la maggior parte del lavoro necessario. Quindi un programma che usa il dispositivo SER: del DOS per gestire le operazioni con la porta seriale non ha bisogno di chiamare la funzione OpenDevice dell'Exec e di inizializzare una struttura IORequest.

I processi dell'Amiga

La Tavola 1 presenta un sommario di tutti i processi che sono generalmente attivi nel sistema Amiga. Per una

Tavola 1: I processi di dispositivo pubblici e privati dell'AmigaDOS

Processi* pubblici *creati dall'utente o dal programmatore (1)

CLI 1	processo di gestione del terminale virtuale CLI 1, RAW, o CON
CLI 2	processo di gestione del terminale virtuale CLI 2, RAW o CON
CLI n	processo di gestione del terminale virtuale CLI n, RAW o CON

Processi pubblici creati dal programmatore

Processo 1	primo processo definito dal programma
Processo 2	secondo processo definito dal programma
Processo n	processo definito dal programma

Processi* privati *creati dal sistema (2)

SYS	processo assegnato al disco di partenza (normalmente DF0)
DF0	processo di gestione file per il disk drive interno, di tipo "DOS" o "KICK" usa direttamente il file trackdisk device
DF1	processo di gestione file per il primo disk drive esterno, di tipo "DOS" o "KICK" usa direttamente il file trackdisk device
DF2	processo di gestione file per il secondo disk drive esterno di tipo "DOS" o "KICK" usa direttamente il file trackdisk device
DF3	processo di gestione file per il terzo disk drive esterno, di tipo"DOS" o "KICK" usa direttamente il file trackdisk device
FFS	processo di gestione file per il Fast-File-System, usa direttamente il file hddisk device
DH0	processo di gestione file per il primo hard disk, usa direttamente il file hddisk device
DH1	processo di gestione file per il secondo hard disk usa direttamente il file hddisk device
AUX	processo di gestione del dispositivo seriale senza buffer usa il file Aux-Handler
SER	processo di gestione del dispositivo seriale con buffer usa il file Port-Handler
PAR	processo di gestione del dispositivo parallelo con buffer, usa il file Port-Handler
PRT	processo di gestione della stampante usa il file Port-Handler
PIPE	processo di gestione dello scambio di input/output tra programmi, usa il file Pipe-Handler
SPEAK	processo di gestione dell'output vocale, usa il file Speak-Handler
CON	processo di gestione del dispositivo console con filtratura dei codici della tastiera, usa il file ConHandler
RAW	processo di gestione del dispositivo console senza filtratura dei codici della tastiera usa il file ConHandler
NEWCON	versione avanzata di CON , usa il file Newcon Handler
RAM	processo di gestione file per la RAM disk non recuperabile, di tipo "DOS" o "KICK
RAD	processo di gestione file per la RAM disk recuperabile, di tipo "DOS o "KICK", usa il file ramdrive device
NIL	processo fittizio che non genera output
disk validator	processo di convalida dei dischi usa il file Disk-Validator
SHELL SEG	processo residente per la gestione delle Shell

(1) I processi logici o di directory definiti dal comando Assign appartengono a questo gruppo
(2) I puntatori alla strutture MsgPort associate alla maggior parte di questi processi vengono ricavati tramite la funzione DeviceProc del DOS

spiegazione più dettagliata, abbiamo diviso questi processi in due categorie principali (processi pubblici e processi privati), anche se dal punto di vista del programmatore questa distinzione è superflua.

Generalmente parlando, i processi completamente pubblici sono processi creati direttamente dai programmatori. In questo caso il programmatore ha il pieno controllo sulle caratteristiche dei processi, in quanto definisce tutti i parametri presenti nella struttura Process e nelle altre strutture relative al processo. Per esempio, può creare un processo CLI, all'interno del suo programma, allocando e inizializzando prima una struttura Process e in seguito una struttura CommandLineInterface (CLI).

I valori specifici usati per la definizione dei parametri delle strutture permettono di controllare in maniera dettagliata lo svolgimento del processo. Oltre a ciò si possono creare dei processi parzialmente pubblici che comunemente corrispondono a dei processi CLI virtuali. In questo caso la creazione e il conseguente svolgimento del processo vengono determinati dal sistema anche se il programmatore può ancora accedere ai parametri che definiscono il processo stesso. Il sistema crea tale processo virtuale automaticamente ogni volta che l'utente o il programmatore apre una finestra CLI.

I processi CLI di gestione del terminale virtuale includono un processo per ogni finestra CLI attualmente aperta, fino a un massimo di 20 finestre CLI aperte contemporaneamente. In generale possiamo dire che la parte di codice definita dal sistema per questi processi parzialmente pubblici non è di solito direttamente accessibile al programmatore. In ogni caso alcune caratteristiche dei più comuni processi attivi possono essere determinate tramite particolari meccanismi di programmazione. Per esempio, il codice di programma eseguibile che apre una finestra CLI, tramite Intuition, e che permette all'utente di ridimensionare in seguito tale finestra, è sempre parte del programma di sistema predefinito per la creazione di processi CLI, anche se l'apertura della finestra è stata provocata dall'utente. Ad ogni modo, se un programmatore scrive un programma che apre un file di testo per visualizzarlo in una qualsiasi finestra CLI attivata precedentemente dall'utente, tale programma può determinare le caratteristiche del processo CLI creato dal sistema che è associato a quella finestra. Il programmatore può quindi controllare e modificare le dimensioni e i colori correnti di tale finestra CLI tramite il proprio programma. La struttura CommandLineInterface associata a ogni processo CLI permette cioè al programmatore di determinare le caratteristiche di ogni processo CLI attivo indifferentemente dalle sue origini.

In generale i processi privati sono quei processi predefiniti generati dal sistema il cui codice eseguibile è stato precedentemente scritto dai programmatori del sistema operativo dell'Amiga. Questi processi vengono attivati automaticamente dal sistema non appena esso riconosce in maniera appropriata tutte le componenti fisiche e logiche dell'hardware. È importante ricordare che il sistema genera un certo numero di questi processi, come elementi standard della propria architettura, durante l'esecuzione del boot.

I processi per i dispositivi DF0: e DF1: vengono generati in modo automatico indipendentemente dalle specifiche presenti nel file Mountlist, le quali determinano invece, tramite il comando Mount, i processi privati che possono essere attivati dopo il boot. In una configurazione tipica, i dispositivi creati dal sistema per la gestione dei file su disco consistono in un massimo di sette processi: DF0:, DF1:, DF2:, DF3:, DH0:, DH1: e FFS: (vedere Tavola 1).

Se nel file Mountlist sono presenti delle specifiche per DF0: o DF1:, e viene impartito il comando Mount DF0: o Mount DF1:, le caratteristiche di input/output dei processi standard di sistema di tali dispositivi possono essere ridefinite. Allo stesso modo, se il file Mountlist contiene delle specifiche per il dispositivo NEWCON: del DOS e viene impartito il comando Mount NEWCON:, il sistema crea un processo privato e cioè alloca e inizializza automaticamente una struttura Process e le relative strutture per quel processo.

I processi privati per la gestione del dispositivo seriale sono di due tipi: con buffer (SER:) e senza buffer (AUX:). PAR: è il processo per la gestione, con buffer, del dispositivo parallelo. PRT: è il processo per la gestione della stampante. Il dispositivo per la gestione della console è rappresentato da tre tipi di processi: CON:, NEWCON: e RAW:. La differenza fra questi tre processi è evidenziata nella Tavola 1.

I processi di gestione del RAM disk sono di due tipi: RAM disk recuperabile (RAD:) e non recuperabile (RAM:). SPEAK: è il processo che fa "parlare" l'Amiga prelevando i dati da un buffer interno e producendo, tramite fonemi, i segnali audio corrispondenti. Infine NIL: è il dispositivo che accetta l'input da qualsiasi sorgente e non produce nessun output così che i dati vengono persi e non c'è modo di accedervi. Provate a confrontare i dispositivi della Tavola 1 con le specifiche del file Mountlist che trovate nei vostri dischi e con i comandi Mount presenti nella Startup-sequence, per notare come il sistema determina le caratteristiche dei dispositivi e dei processi da creare.

Le strutture dell'AmigaDOS

Nella pagina successiva, la Tavola 2 riassume tutte le strutture i cui parametri vengono usati per controllare il sistema di gestione dei processi DOS.

All'inizio della lista abbiamo la struttura DosLibrary, che consiste in una libreria di tipo Exec con l'aggiunta di informazioni specifiche sul sistema DOS. Studiando la struttura Library dell'Exec si può vedere come la struttura DosLibrary permette ai programmi di richiamare le funzioni della libreria DOS.

La struttura RootNode è un'altra importante struttura DOS; contiene dei parametri che permettono di conoscere dettagliatamente tutti i processi CLI correntemente attivi. Più precisamente, la lista di tutti i processi CLI attivi nel sistema è contenuta nel parametro rn_TaskArray di tale struttura. La struttura RootNode definisce inoltre dei parametri relativi a processi generici come il Disk Validator e il File Handler.

La struttura DosInfo è stata definita per una possibile gestione dell'Amiga in ambiente di rete e contiene, oltre alle specifiche relative alla rete, il puntatore a una lista sempre aggiornata dei dispositivi e dei sistemi collegati all'Amiga; questa lista è costituita da una serie di strutture DevInfo concatenate tra loro.

La struttura DevInfo definisce un nodo di una lista concatenata per i tre tipi di dispositivi DOS: reali (o fisici), logici e di volume.

Tavola 2: Sommario delle strutture usate dall'AmigaDOS

DosLibrary (1)	Una struttura Library estesa che contiene informazioni specifiche sul sistema DOS
RootNode	Definisce una lista di segmenti per tutti i processi CLI attivi, per il Disk Validator e per il File Handler generico
DosInfo	Definisce la funzione dell'Amiga all'interno di una rete e contiene il puntatore a una struttura DevInfo
DevInfo	Definisce una lista concatenata per i dispositivi reali, logici e di volume
DeviceNode	Definisce soltanto una lista per i dispositivi reali
DeviceList	Definisce soltanto una lista per i dispositivi di volume
DosEnvec	Definisce le caratteristiche fisiche dei dispositivi contenute nel file MountList
Process (2)	Definisce le caratteristiche dei processi presenti nel sistema (task, message port, stack)
CommandLineInterface	Contiene informazioni su ciascun processo CLI attivo
FileInfoBlock	Contiene informazioni su ciascun file aperto
InfoData (3)	Contiene informazioni sullo stato attuale dei dischi inseriti nei disk drive
FileHandle (4)	Contiene le strutture Message e MsgPort per la gestione dei file aperti
DosPacket	Contiene le strutture Message e MsgPort per i DOS Packet
StandardPacket	Riunisce le strutture DosPacket e Message in una struttura più funzionale
FileSysStartMsg	Definisce il messaggio di partenza per l'apertura dei dispositivi DOS
FileLock (5)	Definisce il nodo di una lista per gli accessi riservati dei file e delle directory
DateStamp (6)	Definisce la data del sistema. Viene usata nelle strutture RootNode, DeviceList, DosList e FileInfoBlock
SegList (7)	Definisce la lista dei segmenti di codice di un processo Riferimenti a tale "struttura" sono presenti nelle strutture RootNode, Process, CommandLineInterface, DevInfo e DosList

(1)	Il puntatore alla struttura Library si ottiene con la funzione OpenLibrary dell'Exec
(2)	Il puntatore alla struttura MsgPort contenuta nella struttura Process si ottiene con le funzioni CreateProc e DeviceProc
(3)	La struttura InfoData viene inizializzata dalla funzione Info
(4)	Il puntatore alla struttura FileHandle si ottiene con le funzioni Open, Input e Output
(5)	Il puntatore alla struttura FileLock si ottiene con le funzioni Lock, Duplock, CreateDir, CurrentDir e ParentDir
(6)	Il puntatore alla struttura DateStamp si ottiene con la funzione DateStamp
(7)	Il puntatore alla "struttura" SegList si ottiene con la funzione LoadSeg

La struttura DeviceNode è molto simile alla struttura DevInfo. Essa ha però uno scopo più limitato; infatti definisce una lista concatenata soltanto per i dispositivi reali.

Anche la struttura DeviceList è simile alla struttura DevInfo, ma definisce soltanto una lista dei dispositivi di volume.

Generalmente, la struttura DosEnvec contiene la descrizione delle specifiche di un dispositivo fisico, per esempio il disco, prelevate da quelle presenti nel file Mountlist. Ciò avviene durante l'inizializzazione del dispositivo tramite il comando Mount.

La struttura Process contiene la maggior parte dei parametri più importanti per la definizione delle caratteristiche di ciascun processo del sistema, indifferentemente dalle sue origini. Essa contiene una struttura Task associata a tale processo, una struttura MsgPort usata per comunicare con gli altri processi, un puntatore allo stack di tale processo e molte altre importanti definizioni.

Tutti i processi CLI sono definiti da una specifica struttura CommandLineInterface. Ciascuna di queste strutture contiene infatti il nome della directory corrente associata a una finestra CLI specifica, il codice di errore generato dall'ultimo comando DOS impartito da tale finestra, i puntatori alle strutture FileHandler dei dispositivi di input/output associati al processo, informazioni sullo stack e altre informazioni necessarie per il funzionamento di un processo CLI. Ricordiamo che la struttura RootNode contiene la lista corrente di tutti i processi CLI attivi nel suo parametro rn_TaskArray.

A ogni file aperto viene associata una struttura FileInfoBlock che ne riflette le caratteristiche. Ricordiamo tra queste il parametro fib_DiskKey che definisce l'attuale posizione del file sul disco.

Le strutture InfoData, una per ogni disco inserito nei disk drive, descrivono le attuali condizioni del disco (per esempio se è protetto da scrittura).

Ciascuna struttura FileHandle contiene informazioni per la getione dei file attualmente aperti. Ricordiamo che le funzioni Open, Input e Output del DOS restituiscono un puntatore a una struttura FileHandle.

Le strutture DosPacket e StandardPacket definiscono il contenuto dei messaggi Exec usati dai processi di programma che intendono comunicare con i processi dell'AmigaDOS per svolgere determinate operazioni sui file, per esempio leggere un certo numero di byte. Entrambe le strutture permettono al sistema di gestire le specifiche richieste DOS generate da programma e di verificare se tali operazioni sono state eseguite con successo.

La struttura FileSysStartMsg viene usata dal sistema per trasmettere le richieste DOS ai dispositivi Exec predisposti per la gestione dei file. Essa infatti, oltre al numero dell'unità e al nome del dispositivo coinvolto, contiene anche i bit di flag che tale dispositivo usa per controllare il tipo di operazione richiesta.

La struttura FileLock definisce un nodo di una lista concatenata che rappresenta gli accessi riservati a un file o una directory di uno specifico disco (o volume). Questo sistema di accesso riservato garantisce la corretta lettura o scrittura dei dati in un file anche quando tale file (o directory) viene usato contemporaneamente da più task o processi. Ricordiamo infine che sei funzioni DOS restituiscono un puntatore a tale struttura.

La struttura DateStamp contiene la data di sistema in giorni, minuti e "tick" (cinquantesimi di secondo) permettendo al sistema di aggiornare la data di creazione dei file descritta all'interno di ognuno di essi. La funzione DateStamp del DOS restituisce un puntato-

re a tale struttura.

La struttura SegList consiste in una lista concatenata di segmenti di codice eseguibile che compongono un programma in memoria. In questo caso la parola "struttura" viene usata impropriamente in quanto non ne è presente una definizione formale nei file di inclusione DOS. Ogni programma eseguibile può essere suddiviso in vari segmenti. Quando le routine interne dell'AmigaDOS caricano un file eseguibile distribuiscono in memoria i segmenti che lo costituiscono. Lo stesso avviene quando un programma usa la funzione LoadSeg del DOS per caricare del codice eseguibile. Pertanto i segmenti eseguibili del programma caricato possono trovarsi in punti diveri della memoria; il sistema usa quindi la SegList associata a tale programma per determinare dove sono state collocate tutte le parti eseguibili in modo che la CPU possa saltare in queste locazioni per eseguire il programma.

La coesistenza dei processi

Come abbiamo già detto, il sistema contiene, per ogni dato momento, svariati processi che potenzialmente interagiscono tra loro. Molti di questi processi sono stati creati dal sistema per consentire al vostro programma e ai processi di altri programmi di accedere ai dati e alle informazioni presenti dovunque nel sistema stesso (per esempio sui dischi, usando il trackdisk.device, o tramite un modem, usando il serial.device).

Possiamo chiarire il concetto d'interazione tra processi con un diagramma. La Figura 3 illustra le relazioni che intercorrono tra il processo creato dal vostro programma e alcuni dei più importanti processi coesistenti nel sistema. Il rettangolo posto al centro sulla parte alta del diagramma rappresenta il processo creato esplicitamente dal vostro programma mediante la dichiarazione di una struttura Process. Notiamo inoltre che tutti i processi, sia pubblici che privati, posseggono una propria message port tramite la quale avviene la comunicazione. Sembra inoltre che le funzioni della libreria DOS diano più importanza a queste message port, e alle relative strutture MsgPort, piuttosto che ai processi stessi e alle relative strutture Process. Questo strano modo di agire risulta evidente quando, tramite una chiamata alla funzione CreateProc o DeviceProc, otteniamo un puntatore a una struttura MsgPort piuttosto che a una struttura Process. Ciò è contrario a quanto ci si aspetta facendo riferimento al nome di queste due funzioni. Di fatto nella libreria Dos non esiste nessuna funzione che restituisca direttamente il puntatore a una struttura Process. Se per esempio vogliamo ottenere il puntatore alla struttura Process del processo DF0:, tramite una chiamata alla funzione DeviceProc, dobbiamo usare la seguente procedura:

```
struct MsgPort *msgPort;
struct Process *process;

msgPort = DeviceProc("DF0:");
process = (struct Process *) (msgPort -
 sizeof(struct Task));
```

Questo è dovuto al fatto che la struttura Process possiede all'interno prima una struttura Task e in seguito una struttura MsgPort. Sottraendo al puntatore della struttura MsgPort il valore corrispondente alla dimensione della struttura Task otteniamo il puntatore alla struttura Task stessa e quindi il puntatore alla struttura Process che la contiene. Come vedremo in seguito, la funzione FindTask dell'Exec consente invece a un programma di ottenere direttamente il puntatore alla struttura Process che gestisce il programma stesso (come verrà illustrato più avanti nel programma

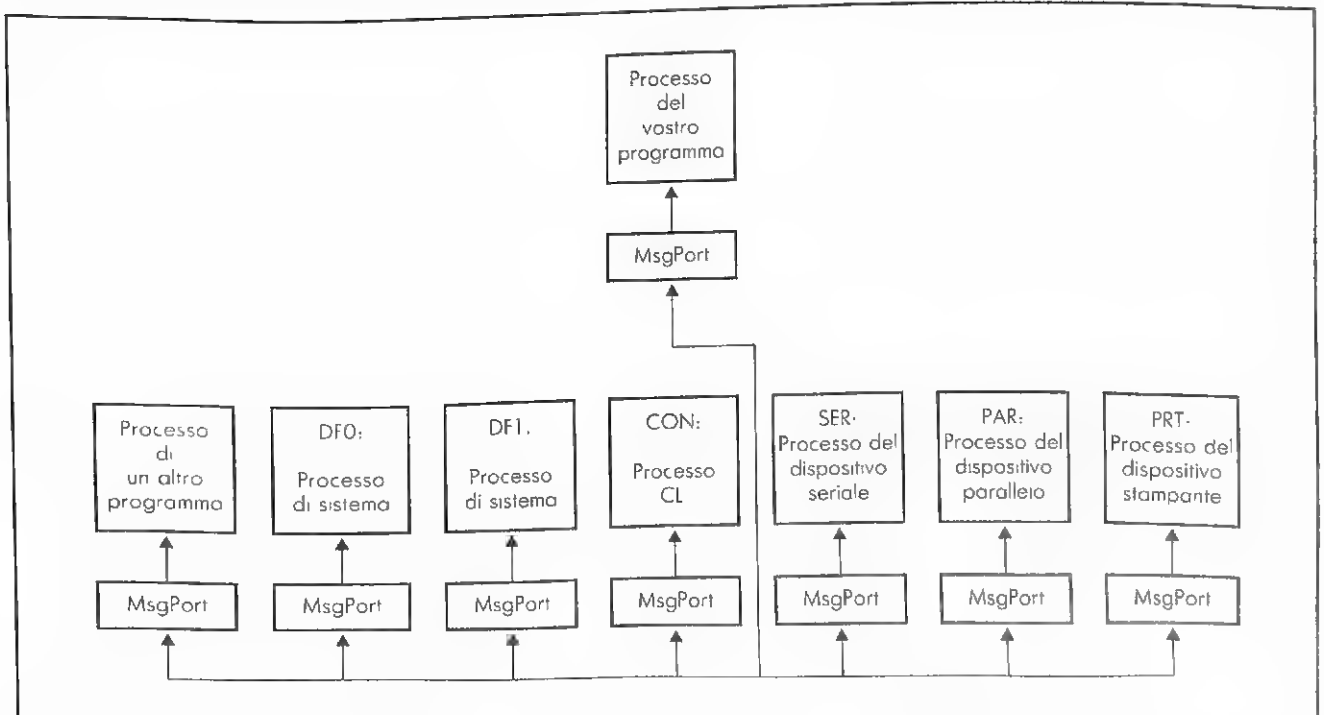



Figura 3: Le relazioni che intercorrono tra processi di programma e di sistema nell'ambito del sistema Amiga

d'esempio). Questa struttura Process è strettamente collegata al processo di gestione della finestra CLI dalla quale è stato mandato in esecuzione il programma.

La diversità che intercorre tra l'AmigaDOS e l'Exec nella gestione delle stesse strutture, è dovuta principalmente al fatto che il sistema AmigaDOS deriva da una conversione diretta di un sistema preesistente, scritto in BCPL, che non gestiva il multitasking. Probabilmente questo sistema usava una struttura Process simile a quella attuale e la struttura Task è stata inserita successivamente durante la conversione per il sistema multitasking. Per rendere il sistema più omogeneo, conservando però le preesistenti funzioni DOS come DeviceProc e CreateProc, è stata in seguito sviluppata la procedura di aggiustamento del puntatore.

La comunicazione mediante le message port

Abbiamo visto che ogni processo può spedire messaggi agli altri processi e può ricevere messaggi di ritorno dalla propria message port. Allo stesso modo il processo creato dal vostro programma può comunicare con gli altri processi attivi nel sistema (Figura 3) tramite la message port che gli è stata assegnata. Il vostro programma ottiene questa assegnazione allocando e inizializzando una struttura Process e quindi la struttura MsgPort in essa contenuta.

Il rettangolo sulla sinistra nella Figura 3 (processo creato da altri programmi) rappresenta dei programmi che sono già stati caricati nel sistema, mentre gli altri rettangoli (processi creati dal sistema) possono essere processi residenti in ROM oppure anch'essi caricati precedentemente. Notiamo per esempio che il vostro programma può comunicare con il processo CON: associato al dispositivo console del CLI, con il processi di gestione file DF0: e DF1:, con il processo del dispositivo seriale SER:, con il processo del dispositivo parallelo PAR:, con il processo di gestione della stampante PRT:, e con molti altri non illustrati nella figura. Tutti i processi, sia di programma che di sistema, necessitano delle strutture Task e MsgPort; i parametri pr_Task e pr_MsgPort delle strutture Process devono quindi essere inizializzati.

Nei processi che fanno uso di dichiarazioni in linguaggio BCPL per definire il proprio codice eseguibile, il parametro pr_GlobVec deve contenere il puntatore a una tabella di vettori a routine BCPL predefinite. Molti processi di sistema, che nelle prime versioni del software (1.1 e 1.2) usavano dichiarazioni in BCPL, sono stati riscritti in C o in Assembly, e quindi nelle strutture Process che li rappresentano il parametro pr_GlobVec non contiene un puntatore ma un valore fittizio (NULL oppure −1). È importante capire che i parametri di una struttura Process devono essere inizializzati a seconda del tipo di processo che tale struttura rappresenta. Per esempio, tutti i processi che non sono processi CLI avranno i parametri pr_TaskNum, pr_CLI, pr_ConsoleTask e pr_WindowPtr impostati a zero. Per tutti i processi di gestione dei file (DF0:, DF1:, DH0:, DH1:...) il parametro pr_FileSystemTask conterrà il puntatore a una struttura MsgPort. Studiate attentamente la definizione di ciascun parametro della struttura Process, che descriveremo ora, per capire la loro funzione a seconda del tipo di processo.

La struttura Process

Nel file include dosextens.h la struttura Process è definita in questo modo:

```
struct Process {
  struct Task pr_Task;
  struct MsgPort pr_MsgPort;
  WORD pr_Pad;
  BPTR pr_SegList;
  LONG pr_StackSize;
  APTR pr_GlobVec;
  LONG pr_TaskNum;
  BPTR pr_StackBase;
  LONG pr_Result2;
  BPTR pr_CurrentDir;
  BPTR pr_CIS;
  BPTR pr_COS;
  APTR pr_ConsoleTask;
  APTR pr_FileSystemTask;
  BPTR pr_CLI;
  APTR pr_ReturnAddr;
  APTR pr_PktWait;
  APTR pr_WindowPtr;
};
```

Esaminiamo in dettaglio il significato dei parametri:

pr_Task – una specifica struttura Task associata al processo. La presenza di questa struttura dimostra che un processo è in effetti un task esteso. Le routine interne del DOS usano questa struttura per gestire lo stack del task associato al processo, per dialogare con i segnali Exec (in particolare il signal bit numero otto) che arrivano alla message port del processo, per definire gli eventi trap del processo e per controllare tutte le caratteristiche specifiche di un task.

pr_MsgPort – una specifica struttura MsgPort assegnata al processo che serve, come abbiamo visto, per la ricezione di messaggi Exec provenienti da altri processi del sistema. In particolare, le routine interne del DOS trasmettono a questa message port dei messaggi sotto forma di DOS Packet, ciascuno caratterizzato da una struttura Message. Il codice eseguibile del processo può interpretare questi messaggi e agire di conseguenza.

pr_Pad – una word di allineamento come richiesto dal linguaggio BCPL.

pr_SegList – un puntatore alla lista concatenata dei segmenti del processo. Questa lista viene usata dal DOS per determinare dove si trova il codice eseguibile del processo dopo che questo è stato caricato e distribuito in memoria.

pr_StackSize – la dimensione in byte dello stack del processo. Il DOS usa questo stack per depositare sia i dati temporanei usati dalle funzioni del processo sia gli indirizzi di ritorno delle chiamate alle proprie funzioni.

pr_GlobVec – il vettore globale usato dal DOS per accedere alle routine private BCPL del processo. Se tale processo non è stato scritto in BCPL o se non usa le routine BCPL del DOS, questo parametro non contiene un indirizzo reale. Per esempio, un processo scritto in puro linguaggio C o Assembly usa il valore −1 per indicare che il processo non è stato scritto in BCPL.

pr_TaskNum – il numero del task CLI associato al processo che normalmente compare nel prompt della specifica finestra CLI. Se il task non è un processo CLI questo parametro contiene il valore zero (NULL).

pr_StackBase – un puntatore al limite alto dello stack del processo. Spesso questo valore coincide con quello definito all'interno della struttura Task.

pr_Result2 – il corrente valore di ritorno proveniente dall'ultima chiamata di funzione effettuata dal processo che coincide con il valore restituito dalla funzione IoErr del DOS. Il vostro programma può usare questo valore per determinare l'esito delle chiamate alle funzioni DOS.

pr_CurrentDir – un puntatore alla lista di percorsi di directory (path) associata alla directory corrente. Serve per determinare la directory o il percorso di directory da cui è stato mandato in esecuzione il programma.

pr_CIS – un puntatore alla struttura FileHandle associata al flusso di input corrente. Questo puntatore viene usato per determinare il file di input attualmente assegnato al processo. Nel caso di un processo CLI l'input avviene normalmente tramite la tastiera (stdin).

pr_COS – un puntatore alla struttura FileHandle associata al flusso di output corrente. Questo puntatore viene usato per determinare il file di output attualmente assegnato al processo. Nel caso di un processo CLI l'output viene normalmente diretto verso lo schermo (stdout).

pr_ConsoleTask – un puntatore alla struttura MsgPort del processo di gestione del dispositivo console. Se il nostro processo non è associato a un processo CLI di gestione della console, input da tastiera e output su video, questo puntatore contiene il valore zero (NULL).

pr_FileSystemTask – un puntatore alla struttura MsgPort del processo di gestione dei file. Se il nostro processo non è associato a un processo di gestione dei file questo puntatore contiene il valore zero (NULL).

pr_CLI – un puntatore alla struttura CommandLineInterface associata al processo. Se il nostro processo non è un processo CLI questo puntatore contiene il valore zero (NULL).

pr_ReturnAddr – un puntatore all'indirizzo di ritorno, contenuto nello stack del processo, dell'ultima chiamata a una funzione. Normalmente questo valore non viene usato dai programmi utente.

pr_PktWait – un puntatore alla funzione che viene chiamata durante l'attesa di un messaggio Exec di ritorno al processo. Il processo attende l'arrivo di un messaggio mediante la funzione WaitPort dell'Exec. Quando un messaggio arriva alla message port del processo, la funzione GetMsg dell'Exec intercetta l'attivazione del signal bit numero otto e passa il controllo a questa funzione. Se per esempio volete che il vostro programma esegua delle specifiche operazioni ogni volta che una chiamata alla funzione Open del DOS fallisce, dovrete mettere in questo parametro il puntatore alla funzione del vostro programma che controlla il parametro pr_Result2 e, in base al valore in esso contenuto, svolge le operazioni richieste.

pr_WindowPtr – un puntatore alla struttura Window di Intuition usata per visualizzare uno specifico requester richiesto dal processo. Se per esempio il processo richiede che un disco o volume specifico sia presente in un disk drive, farà apparire automaticamemte un requester di volume nella parte in alto a sinistra dello schermo. Tramite questo puntatore si può inoltre accedere a tutti i parametri della struttura Window associata al requester (dimensioni, posizione...).

Le strutture DOS di controllo del sistema

Tra tutte le strutture DOS per la gestione dei processi di cui abbiamo già parlato, quattro in particolare permettono al DOS di avere una visione completa del sistema e cioè la struttura DosLibrary, la struttura RootNode, la struttura DosInfo e la struttura DevInfo. Queste quattro strutture sono collegate tra loro come illustrato nella Figura 4 (a pagina 69). Il programma d'esempio in fondo all'articolo mostra come sfruttare questo collegamento per ottenere informazioni su tutti i dispositivi presenti nel sistema.

La struttura DosLibrary è la struttura fondamentale del sistema DOS e viene definita nel file include dosextens.h come segue:

```
struct DosLibrary {
  struct Library dl_lib;
  APTR dl_Root
  APTR dl_GV
  LONG dl_A2;
  LONG dl_A5;
  LONG dl_A6
}
```

Esaminiamo in dettaglio il significato dei parametri:

dl_lib – la struttura Library che rappresenta un nodo della lista delle librerie gestite dall'Exec. Questa struttura viene usata per accedere a tutte le funzioni della dos.library, Input, Output, CreateDir..., di cui abbiamo discusso in un precedente articolo. Come per le altre librerie dell'Exec, tutti i vettori di chiamata alle funzioni (jump vector) sono posizionati a offset negativi rispetto al puntatore della libreria. Il puntatore a questa struttura è ottenuto con la funzione OpenLibrary dell'Exec.

dl_Root – un puntatore a una specifica struttura RootNode, che vedremo in seguito.

dl_GV – un puntatore al Global Vector BCPL che consiste in una tabella di salto (jump table) usata dalle routine interne del DOS per accedere alle funzioni, generalmente private, dell'AmigaDOS scritte in BCPL. I nomi di queste funzioni BCPL non sono di pubblico dominio anche se alcuni di essi sono presenti nelle ultime informazioni provenienti dai programmatori, come per esempio loaddevice, finddevice e sendpacket.

dl_A2, dl_A5, dl_A6 – puntatori privati usati dal DOS che contengono i valori correnti dei rispettivi registri della CPU. Il loro scopo non è importante nel contesto di questo articolo.

Abbiamo visto che la struttura DosLibrary è collegata alla struttura RootNode tramite il puntatore contenuto in dl_Root. Questa struttura RootNode è spesso citata come struttura generale in quanto contiene le informazioni associate ai più importanti processi del sistema. La definizione di questa struttura, anch'essa contenuta nel file include dosextens.h, appare come segue:

```
struct RootNode {
  BPTR rn_TaskArray;
  BPTR rn_ConsoleSegment;
  struct DateStamp rn_Time;
  LONG rn_RestartSeg;
  BPTR rn_Info;
  BPTR rn_FileHandlerSegment;
}
```

rn_TaskArray – un BPTR (puntatore BCPL) a un array di puntatori a strutture MsgPort per ciascun processo CLI correntemente attivo nel sistema. Il primo elemento dell'array, TaskTable[0], contiene il numero delle finestre CLI aperte in un dato istante (il valore massimo permesso dal sistema è di 20 processi CLI contemporaneamente attivi). Il secondo elemento, TaskTable[1], è un puntatore alla struttura MsgPort che rappresenta il processo CLI numero 1. Il terzo elemento è un puntatore alla struttura MsgPort che rappresenta il processo CLI numero 2. L'ennesimo elemento, TaskTable[n], è un puntatore alla struttura MsgPort che rappresenta il processo CLI numero n.

A questo punto avrete notato qualcosa di insolito; la terminologia usata dall'AmigaDOS per i riferimenti alle strutture MsgPort è diversa da quella convenzionalmente usata per l'Amiga. Per esempio, anche se nel nome del parametro rn_TaskArray compare la parola "Task", esso è di fatto un puntatore a delle strutture MsgPort. Com'era già stato notato prima, a proposito dei puntatori alle strutture Process, questa è una convenzione tipica dell'AmigaDOS e bisogna stare attenti a interpretare tutti i riferimenti ai parametri di strutture che contengono le parola "task" o "Task" come riferimenti diretti o indiretti a strutture MsgPort piuttosto che a strutture Task. Per esempio, nella struttura Process il puntatore alla struttura MsgPort del processo del dispositivo console appare con il nome pr_ConsoleTask e il puntatore alla struttura MsgPort del processo di gestione dei file appare con il nome pr_FileSystemTask.

rn_ConsoleSegment – un BPTR alla lista dei segmenti di codice per un generico processo CLI. Viene usato dalle routine interne del DOS per accedere al codice di creazione di un processo CLI ogni volta che l'utente o il programmatore impartisce il comando NEWCLI o RUN. L'AmigaDOS usa quindi una sola copia del codice per creare tutte le finestre CLI. In questo caso, basandovi sulla vostra conoscenza di Intuition, potete immaginare che tale codice contenga l'allocazione e l'inizializzazione di una struttura NewWindow, l'apertura della finestra mediante una chiamata alla funzione OpenWindow di Intuition e il codice richiesto per collegare tale finestra all'unità del dispositivo console.

rn_Time – una specifica struttura DateStamp che contiene la data corrente del sistema continuamente aggiornata dall'AmigaDOS.

rn_RestartSegment – un BPTR alla lista dei segmenti del processo di convalida dei dischi (Disk Validator) usato dall'AmigaDOS quando un disco viene inserito nel disk drive. Ricordiamo che il file Disk-Validator dev'essere presente nella directory L del disco sistema.

rn_Info – un BPTR a una struttura DosInfo usata per la definizione del funzionamento di un Amiga all'interno di una rete, come vedremo più avanti.

rn_FileHandlerSegment – un BPTR alla lista dei segmenti di codice generico per la gestione dei file usato dal DOS per gestire tutti i processi del sistema che fanno uso di file.

Le informazioni contenute in una struttura DosInfo hanno la loro importanza soltanto nel caso in cui l'Amiga faccia parte di una rete. In caso contrario la maggior parte dei parametri di tale struttura saranno posti a zero (NULL). La definizione della struttura DosInfo che appare nel file include dosextens.h è la seguente:

```
struct DosInfo {
  BPTR di_McName;
  BPTR di_DevInfo;
  BPTR di_Devices;
  BPTR di_Handlers;
  BPTR di_NextEntry;
  LONG di_UseCount;
  BPTR di_SegPtr;
  BPTR di_SegName;
}
```

Prima di passare alla descrizione dei parametri ricordiamo che questa struttura ha subito delle modifiche nella release software 1.2. Gli ultimi quattro parametri sostituiscono il parametro di_Nethand che era presente nella release software 1.1.

di_McName – un BPTR al nome della rete di cui questo specifico computer Amiga rappresenta un nodo. Se il vostro Amiga non fa parte di una rete questo valore è impostato a zero (NULL).

di_DevInfo – un BPTR a una specifica struttura DevInfo. Un gruppo di strutture DevInfo rappresenta una lista concatenata di tutti i dispositivi reali, logici e di volume presenti nel sistema.

di_Devices, di_Handlers – il significato originale di questi parametri non è molto chiaro. Ad ogni modo se il vostro Amiga non fa parte di una rete questi valori sono impostati a zero.

Come già accennato, i prossimi quattro parametri sono stati aggiunti nella release software 1.2 e servono per definire la lista dei segmenti residenti usati internamente dal DOS. La documentazione attualmente disponibile non ci consente di avere un'idea chiara di come questa lista venga usata.

di_NextEntry – un BPTR all'elemento successivo nella lista dei segmenti residenti. Potete usare questo puntatore per trovare tutti gli altri segmenti residenti del processo.

di_UseCount – un contatore che viene incrementato ogni volta che un processo usa la lista dei segmenti residenti. Viene impostato a –1 nel caso in cui questo specifico segmento non sia disponibile.

di_SegPtr – un BPTR a uno specifico segmento della lista.

di_SegName – un BPTR al nome di uno specifico segmento della lista.

Ogni dispositivo presente nel sistema è rappresentato da una struttura DevInfo e l'insieme di queste strutture forma una lista concatenata. Per capire questo meccanismo osservate come il programma d'esempio scorre questa lista e stampa le informazioni che riguardano ciascun dispositivo.

Anche la seguente definizione appare nel file include dosextens.h:

```
struct DevInfo {
  BPTR dvi_Next;
  LONG dvi_Type;
  APTR dvi_Task;
  BPTR dvi_Lock;
  BSTR dvi_Handler;
  LONG dvi_StackSize;
  LONG dvi_Priority;
  LONG dvi_Startup;
  BPTR dvi_SegList;
  BPTR dvi_GlobVec;
```

```
  BSTR dvi_Name;
};
```

dvi_Next – un BPTR alla prossima struttura DevInfo della lista. Se questa è l'ultima struttura della lista contiene il valore zero (NULL).

dvi_Type – il tipo specifico di dispositivo rappresentato da questa struttura DevInfo.

I tipi di dispositivo sono definiti come segue:

dvi_Type = DLT_DEVICE	(0) rappresenta un dispositivo DOS reale;
dvi_Type = DLT_DIRECTORY	(1) rappresenta un dispositivo DOS logico;
dvi_Type = DLT_VOLUME	(2) rappresenta un dispositivo di volume.

dvi_Task – un puntatore alla struttura MsgPort assegnata al processo che gestisce questo dispositivo. Se il processo non possiede una message port questo valore è nullo. Anche in questo caso la parola Task si riferisce a una struttura MsgPort.

dvi_Lock – un BPTR alla struttura FileLock che rappresenta un file o una directory con accesso riservato a questo dispositivo. Se il dispositivo non sta usando un file o una directory questo valore è nullo. Ricordiamo che alcune funzioni del DOS, tra cui Lock e Duplock, restituiscono il puntatore a una struttura FileLock.

dvi_Handler – un BSTR (puntatore a una stringa BCPL) al nome del file predisposto per la gestione del dispositivo (device handler). Nel caso del dispositivo RAM: questo parametro punterà al nome RAM-Handler. Il relativo file dev'essere presente nella directory L del disco sistema. Le routine interne dell'AmigaDOS usano questo valore per determinare dove si trova il codice eseguibile associato a uno specifico dispositivo che richiede un device handler. Se il puntatore è impostato a zero (NULL) significa che il dispositivo DOS comunica direttamente col sottostante dispositivo Exec e non necessita di un device handler.

dvi_StackSize – una longword che rappresenta la dimensione dello stack del device handler. Se il dispositivo richiede un device handler questo valore deve essere inizializzato. Per i dispositivi logici e di volume generalmente viene posto a 0L.

dvi_Priority – una longword che rappresenta la priorità di esecuzione del task per il processo handler di questa specifica struttura DevInfo.

dvi_Startup – una longword che rappresenta il valore di partenza da passare al processo handler di questa specifica struttura DevInfo. Le informazioni attualmente disponibili non spiegano chiaramente come questo valore viene usato dalle routine interne del DOS.

dvi_SegList – un BPTR alla lista dei segmenti di codice eseguibile per questo processo handler.

dvi_GlobVec – un BPTR al vettore globale delle routine interne dell'AmigaDOS. Se questo processo è stato scritto in puro linguaggio C o Assembly, cioè non possiede variabili globali BCPL, tale parametro contiene il valore –1. Viene invece impostato a zero per indicare che il codice del processo segue le regole del linguaggio BCPL e che l'AmigaDOS deve creare un vettore globale per questo processo.

dvi_Name – un BSTR al nome del dispositivo rappresentato da questa struttura DevInfo.

Commento al listato

Il programma d'esempio vi permette di determinare come questi parametri sono impostati nel vostro sistema. In questo programma vengono aperte due librerie: la DosLibrary e l'ArpLibrary. Facciamo notare che, per una corretta compilazione del programma, i file d'inclusione Arp, libraries/arpbase.h e libraries/arpfunc.h, devono essere accessibili al compilatore nel path di ricerca dei file di inclusione, come pure il file arp.library deve essere presente nella directory LIBS:. La libreria Arp viene usata dal programma solo per le conversioni di stringhe BSTR in stringhe C. Nel caso in cui il file arp.library, di cui viene richiesta la versione 34, non fosse accessibile al programma, apparirà sullo schermo un messaggio di alert. Un fallimento durante l'apertura della dos.library provocherà invece l'uscita dal programma con un messaggio che ne spiega la ragione.

Dopo l'apertura delle librerie il programma procede con la stampa del parametro dl_GV contenuto nella struttura DosLibrary. Vengono inoltre ricavati e stampati i puntatori alle strutture RootNode, DosInfo e DevInfo, collegate tra loro come visualizzato nella Figura 4, usando la macro BADDR per convertire i parametri rn_Info e di_DevInfo (puntatori BPTR) in normali puntatori C.

È importante notare che, dal momento che l'Amiga possiede un solo indirizzo fisso, questi indirizzi possono

Figura 4: Relazioni tra le strutture di controllo del DOS

cambiare ogni volta che viene mandato in esecuzione il programma, soprattutto se nel frattempo viene cambiato il contenuto della memoria. La parte successiva del programma consiste in un loop che scorre la lista delle strutture DevInfo e stampa le informazioni riguardanti ciascun dispositivo reale, logico o di volume presente nel sistema. La funzione BtoCStr dell'arp.library viene usata all'interno del loop per convertire i parametri dvi_Name e dvi_Handler (stringhe BSTR) in normali stringhe C. A tale scopo viene definito un buffer chiamato cString nel quale vengono poste le stringhe dopo la conversione. Le successive funzioni printf si riferiscono direttamente al contenuto di questo buffer. Ricordiamo inoltre che la funzione BtoCStr ritorna il numero di caratteri della risultante stringa C.

In caso di stringa nulla verranno stampati messaggi appropriati appropriati per ciascun parametro. Usando inoltre lo stesso buffer per la conversione di entrambe i parametri, il primo carattere di cString viene inizializzato a zero prima di procedere alla conversione. Sappiamo infatti che ciascun dispositivo possiede un nome ma non tutti possiedono un device handler. In seguito alla stampa dei nomi verrà stampato il tipo di dispositivo, la sua priorità, il valore di partenza, la dimensione dello stack, il puntatore alla lista dei segmenti e il vettore globale. Il risultato di queste operazioni di stampa è una lunga lista delle specifiche dei dispositivi presenti nel sistema fra le quali potrete notare per esempio quelle dei dispositivi reali DF0:, DF1:, RAM:, SER:, PAR:, PRT:, CON: e RAW:, quelle dei dispositivi logici SYS:, ENV:, S:, L:, C:, FONTS:, DEVS: e LIBS:, e quelle assegnate ai dischi (dispositivi di volume) correntemente inseriti nei disk drive. L'ultima parte del programma riguarda il processo associato al CLI corrente. Per ottenere informazioni su tale processo abbiamo bisogno del puntatore alla relativa struttura Process. Sappiamo inoltre che la prima entrata di tale struttura è una struttura Task. Chiamiamo quindi la funzione FindTask dell'Exec con un argomento nullo e convertiamo il puntatore alla struttura Task che ci viene restituito in un puntatore alla struttura Process.

Ricordiamo che chiamando la funzione FindTask con un argomento nullo si ottiene il puntatore al task corrente, cioè quello associato al programma stesso. Questa è una procedura standard dell'Amiga per ottenere il puntatore alla struttura Process appartenente a uno specifico processo CLI. Facciamo comunque notare che, mentre tutti i processi hanno un task associato, non tutti i task fanno parte di un processo. A questo punto possiamo accedere a tutti i valori contenuti nella struttura Process del CLI. Viene quindi stampato il numero di task della finestra CLI corrente, il vettore globale BCPL, che corrisponderà a quello della struttura DosLibrary, il puntatore alla lista dei segmenti di codice del processo, il puntatore alla message port, il puntatore allo stack e la sua dimensione. Vengono infine usate le informazioni che riguardano la struttura Task del processo. In questo caso la variabile task1 rappresenta la struttura stessa e non un suo puntatore. L'indirizzo del task viene quindi determinato con &task1 e i valori dello stack con task1.tc_SPUpper e task1.tc_SPLower. Come ultimo valore viene stampato il puntatore alla struttura CommandLineInterface associata al processo CLI attivo.

Conclusioni

In questo articolo abbiamo trattato la struttura Process e le altre quattro importanti strutture che mantengono il controllo del complesso sistema di comunicazione tra i processi dell'Amiga. L'obiettivo era quello di descrivere com'è costruito, e continuamente aggiornato, il sistema multiprocessi dell'Amiga e come avviene la comunicazione tra processi all'interno di questo sistema.

Nel prossimo articolo approfondiremo la conoscenza della struttura CommandLineInterface e dei processi CLI, facendo riferimento a tutte le informazioni contenute in questo articolo. Questo vi permetterà di avere un'ampia e chiara visione dei meccanismi che avvengono durante la gestione di tali processi e servirà per gettare le basi di un argomento più complesso che tratteremo in seguito: i processi di gestione dei file.

```
/*
Process.c
*/

#include <exec/types.h>
#include <exec/execbase.h>
#include <exec/memory.h>
#include <exec/libraries.h>

#include <intuition/intuition.h>

#include <libraries/dos.h>
#include <libraries/dosextens.h>

#include <stdio.h>

#include [illegible]
#include [illegible]

[illegible]

[illegible]

[illegible]
[illegible]

struct DosInfo    *dosInfo    = NULL
struct DevInfo    *devInfo    = NULL

struct Process *process1 = NULL

struct Task task1

struct MsgPort *msgPort1 = NULL

struct CommandLineInterface *CLI = NULL

#define AZTEC_C

/******************************************/

main(argc, argv)
int argc
char *argv[]
{
   char cString[100]

   [illegible]
     [illegible]
       [illegible]
```

LA GESTIONE DEL JOYSTICK IN C

Il dispositivo gameport è la vostra finestra sul joystick

di David T. McClellan

Mentre per gli amanti dei videogame il controllo del joystick facilita le manovre di volo o di combattimento, per i programmatori in C controllare questa periferica risulta un po' più complesso.

I programmi comunicano con il mouse e il joystick tramite il dispositivo gameport, che legge gli eventi di movimento o di pressione dei tasti e li trasmette sotto forma di messaggio al vostro programma. Per comunicare con il dispositivo gameport, dovete innanzitutto creare una message port e inizializzare una struttura IOStdReq, usandole in seguito per aprire il dispositivo. Prendiamo in considerazione il seguente frammento di codice che dev'essere collocato dopo le specificazioni dei file include (exec/types.h, exec/devices.h, exec/io.h, devices/gameport.h e devices/inputevent.h).

```
#define JPT 1
 /* La seconda porta giochi (joystick)
   e' l'unita' 1, la porta mouse e' l'unita' 0      */

struct MsgPort *joyport;
struct IOStdReq *joymsg;
 /* Altre eventuali dichiarazioni
   vanno inserite qui */

if ((joyport = CreatePort(NULL, 0)) == NULL)
 /* Segue il codice che segnala la mancata creazione
   della porta */

else if ((joymsg = CreateStdIO(joyport)) == NULL)
     /* Segue il codice che segnala
        la mancata allocazione della                     */
      /* struttura IOStdReq                              */

     else if (OpenDevice("gameport.device",JPT,joymsg,
       0)) != 0)
          /* Segue il codice che segnala la mancata
             apertura */
          /* del dispositivo                             */
```

Questa parte di codice apre il dispositivo gameport e inizializza joyport e joymsg per poterci comunicare. Facciamo comunque notare che anche se nell'esempio il programma ha aperto il dispositivo per accedere all'unità 1, tale accesso è sempre condiviso con gli altri task, cioè gli eventi provenienti dalla porta giochi potrebbero essere intercettati da altri task con priorità maggiore. Se per esempio state comunicando con la porta mouse mentre Intuition è attivo, esso intercetterà la maggior parte dei messaggi provenienti da tale porta.

Dopo l'apertura del dispositivo, bisogna informare il dispositivo gameport se sta comunicando con un mouse o con un joystick. In particolare per il joystick esistono due tipi diversi di formato dei messaggi: assoluto e relativo. Il formato assoluto invia un messaggio solo quando il joystick cambia posizione; perciò fornisce informazioni precise ma discontinue. Il formato relativo invece invia continuamente dei messaggi, a intervalli regolari, se il joystick non è fermo nella posizione centrale. Il tipo di formato viene definito mediante la trasmissione di un messaggio GPD_SETCTYPE al dispositivo gameport:

```
BYTE buffer[100];
/* . . . altre specifiche del programma */

buffer[0] = GPCT_RELJOYSTICK;
 /* oppure GPCT_ABSJOYSTICK */
joymsg->io_Command = GPD_SETCTYPE;
joymsg->io_Length = 1;
joymsg->io_Data = (APTR) buffer;
SendIO(joymsg);
WaitPort(joyport);
GetMsg(joyport);
/* Per controllare eventuali errori */
```

Il dito sul tasto

Dopo la definizione del formato bisogna specificare quali operazioni devono generare un evento di input, per esempio la pressione e il rilascio del tasto di fuoco e i movimenti orizzontali e verticali del joystick. Possiamo inoltre attivare la generazione automatica di messaggi di timeout nel caso in cui il joystick non venga mosso dall'utente per un determinato periodo di tempo. Queste specifiche vengono passate al dispositivo gameport mediante una struttura GamePortTrigger:

```
struct GamePortTrigger
{
 UWORD gpt_Keys    /* Tasto premuto o rilasciato */
 UWORD gpt_Timeout   /* periodo di timeout
                      in sessantesimi di secondo */
                    /* 0 = messaggi continui */
 UWORD gpt_XDelta  /* Per il joystick
                      va impostato a 1 */
 UWORD gpt_YDelta   /* Per il joystick
                      va impostato a 1 */
}
```

Il primo parametro, gpt_Keys, viene impostato a (GPTF_UPKEYS GPTF_DOWNKEYS) per gestire sia la pressione che il rilascio del tasto. Nel secondo, gpt_Timeout, viene specificato, in sessantesimi di secondo, il periodo di attesa di un evento prima della generazione automatica del messaggio di timeout. Gli ultimi due parametri vengono usati soprattutto per il mouse e servono per definire lo spostamento richiesto per la generazione di messaggi relativi al movimento orizzontale e verticale. Per informare il dispositivo gameport di queste scelte, è sufficiente inviare un altro messaggio tramite joymsg, contenente in io_Data il puntatore alla struttura GamePortTrigger, in io_Length la lunghezza in byte di tale struttura e in io_Command il valore GPD_SETTRIGGER.

A questo punto siamo pronti a ricevere gli eventi di input che ci arriveranno tramite una struttura InputEvent (descritta nel file include devices/inputevent.h). Ecco un esempio di come leggere e decodificare i messaggi del joystick:

```
struct InputEvent joyevent;
int code, dx,dy;
/* ... altre specifiche del programma ... */

joymsg->io_Command = GPD_READEVENT;
joymsg->io_Data = (APTR) &joyevent;
joymsg->io_Length = sizeof(struct InputEvent);
joymsg->io_Flags = 0; /* Per evitare il Quick I/O */

for (;;) /* ciclo continuo */
{
 SendIO(joymsg);
 WaitPort(joyport);
 GetMsg(joyport);
 code = joyevent.ie_Code;
 dx = joyevent.ie_X;
 dy = joyevent.ie_Y;
 if(code == IECODE_LBUTTON)
 /* pressione del tasto */
 else if(code == (IECODE_UP_PREFIX | IECODE_LBUTTON))
      /* rilascio del tasto */
 if(dx > 0)
 /* movimento verso destra */
 else if (dx < 0)
      /* movimento verso sinistra */
 if(dy > 0)
 /* movimento verso l'alto */
 else if(dy < 0)
      /* movimento verso il basso */
 /* ... altro codice per la gestione del joystick */
 /* ed eventuale uscita dal loop ... */
};
```

A ogni ripetizione del ciclo viene letto un evento e interpretato. Facciamo comunque notare che durante l'attesa dell'evento non viene interrotto il multitasking. Gli eventi relativi al tasto, al movimento orizzontale e al movimento verticale sono distinti tra loro, ma possono essere presenti contemporaneamente nello stesso messaggio. Quando abbiamo finito di comunicare con il joystick ripristiniamo il tipo di controllo con un comando GPD_SETCTYPE, allo stesso modo usato prima per definire il formato relativo, usando in questo caso il valore GPCT_NOCONTROLLER. Infine chiudiamo il dispositivo gameport con CloseDevice(), rimuoviamo joymsg con DeleteStdIO() e joyport con DeletePort().

Con un po' di pratica, l'uso di queste routine vi apparirà facile come abbattere i Cylon (vi ricordate della serie *Galactica*?). ■

POWERWINDOWS 2.5: LA PROGRAMMAZIONE AUTOMATICA

Eccezionale: da oggi non è più necessario pensare alla stesura di codici per la parte grafica dei vostri programmi. Ci pensa infatti questa creatura della Inovatronics

di Eugene P. Mortimore

In precedenti articoli sulla programmazione in C abbiamo visto come costruire un'interfaccia grafica in stile Intuition. Avrete notato che l'uso delle risorse grafiche di Intuition (schermi, finestre, menu...) comporta una lunga serie di definizioni delle strutture appropriate e che ciò richiede al programmatore una grande quantità di tempo e di fatica. Se il vostro obiettivo è quello di produrre un lungo e complesso programma in stile Intuition, sicuramente dovrete affrontare un lungo lavoro di programmazione che comporterà parecchie fasi di stesura, compilazione e correzione prima di ottenere il risultato prefissato.

Quando l'Amiga uscì sul mercato, alcune persone molto ingegnose cominciarono così a pensare che un programma grafico interattivo poteva essere usato per definire altri programmi grafici interattivi. Se un programma dell'Amiga può essere scritto, con un qualsiasi linguaggio compilato, per creare e presentare schermi, finestre, menu e altri oggetti sullo schermo, si può anche farlo funzionare in maniera inversa per creare sorgenti in C (strutture e altre definizioni di supporto), partendo dalle informazioni grafiche inserite sullo schermo dall'utente. Con un simile programma l'utente potrebbe quindi disegnare la propria interfaccia grafica direttamente sullo schermo e lasciare al programma stesso la maggior parte del lavoro più noioso, che consiste nella creazione e nella stesura in linguaggio C di tutte le definizioni di tale interfaccia. Per soddisfare questa esigenza è nato *PowerWindows*.

Il programma

In parole semplici, *PowerWindows* è un programma per la creazione di sorgenti che facilita il programmatore nella stesura dei suoi programmi evitandogli di dover scrivere direttamente e dettagliatamente molte definizioni richieste da uno specifico linguaggio.

Molti altri programmi per la generazione di sorgenti permettono soltanto di usare delle abbreviazioni o dei particolari dialetti di programmazione. Va comunque notato che in questi casi il programmatore deve necessariamente usare il proprio editor di testi e che il testo dev'essere in seguito convertito nel linguaggio desiderato. Il punto di forza di *PowerWindows* sta invece nella possibilità di generare direttamente dei sorgenti in uno specifico linguaggio partendo da informazioni grafiche presenti sullo schermo. Se per esempio volete scrivere un editor di testi simile a *TxED* non avrete bisogno di perdere molto tempo nella definizione dello schermo e dei menu da usare, dal momento che potrete crearli direttamente con *PowerWindows* o meglio ancora potrete prelevarli dallo stesso *TxED* (se è stato precedentemente mandato in esecuzione), e usarli come punto di partenza per successive modifiche a seconda delle proprie preferenze. Usare delle interfacce grafiche già costruite da altri è un sistema molto conveniente e vi permette di concentrare i vostri sforzi sul corpo principale del programma che state scrivendo.

PowerWindows, nella sua versione più recente, può generare sorgenti specifici per i seguenti linguaggi: Aztec C, Lattice C, Assembly, Amiga BASIC, True BASIC, CSI Multi-Forth, TDI Modula-2 e Benchmark Modula-2. Inoltre, sono attualmente in preparazione dei moduli per i linguaggi AC/Fortran, JForth e M2Amiga.

Nella Tavola 1 potete vedere l'intero contenuto del disco di *PowerWindows*. Esso contiene una directory per ogni specifico linguaggio, una directory per gli esempi e tre file eseguibili: PW2.5, PW25Demo e Convert. Inoltre sono presenti dei file di testo, README e CHANGES, che contengono informazioni specifiche sull'attuale versione 2.5 del programma. Gli otto file con l'estensione ".pw" contenuti nella directory Examples illustrano varie importanti caratteristiche di *PowerWindows*. Questi file non sono né file di testo né sorgenti, ma

Tavola 1: Sommario dei file contenuti nel disco di PowerWindows 2.5

Dir.	File	Descrizione
Root		
	PW2.5	File eseguibile del programma PowerWindows 2.5
	Convert	File eseguibile per la conversione dei file oggetto versione 1.x in file oggetto versione 2.5
	PW2.5demo	Versione dimostrativa del programma
	CHANGES	Sommario delle ultime modifiche effettate nella versione 2.5
	README	Descrive alcuni accorgimenti per l'uso del programma Convert
C		
	Aztec example.c	Sorgente d'esempio in linguaggio Aztec C prodotto da PowerWindows
	Lattice example.c	Sorgente d'esempio in linguaggio Lattice C
	README	Descrive l'uso dei sorgenti prodotti da PowerWindows per i compilatori Lattice e Manx
Assembler		
	example.asm	Sorgente d'esempio in linguaggio Assembly
	README	Descrive l'uso dei sorgenti per l'Assembler
AB		
	PW2AB	Programma per la conversione dei file oggetto prodotti da PowerWindows in sorgenti Amiga BASIC
	PW2AB header	File d'intestazione per l'Amiga BASIC
	exec.bmap	File usato per la gestione della libreria Exec
	dos.bmap	File usato per la gestione della libreria Dos
	graphics.bmap	File usato per la gestione della libreria Graphics
	intuition.bmap	File usato per la gestione della libreria Intuition
	README	Descrive l'uso dei sorgenti per l'Amiga BASIC
BM2		
	PW2BM2	Programma per la conversione dei file oggetto prodotti da PowerWindows in sorgenti BenchMark Modula 2
	PW2BM2 header	File d'intestazione per il BenchMark Modula 2
	example.mod	Sorgente d'esempio in linguaggio BenchMark Modula 2
	M2PWChip	File eseguibile per il trasferimento di immagini in memoria chip
	IT1Function.def	File di supporto per il sistema BenchMark Modula 2
	IT1Function.mod	File di supporto per il sistema BenchMark Modula 2
	IT1Function.OBM	File di supporto per il sistema BenchMark Modula 2
	IT1Function.SBM	File di supporto per il sistema BenchMark Modula 2
	PW2.def	File di supporto per il sistema BenchMark Modula 2
	PW2.mod	File di supporto per il sistema BenchMark Modula 2
	PW2.OBM	File di supporto per il sistema BenchMark Modula 2
	PW2.SBM	File di supporto per il sistema BenchMark Modula 2
	README	Descrive l'uso dei sorgenti per il BenchMark Modula 2
MF		
	PW2MF	Programma per la conversione dei file oggetto prodotti da PowerWindows in sorgenti Multi-Forth
	example.mf	Sorgente d'esempio in linguaggio Multi-Forth
	README	Descrive l'uso dei sorgenti per il Multi-Forth
TB		
	PW2TB	Programma per la conversione dei file oggetto prodotti da PowerWindows in sorgenti True BASIC
	PW2TB header	File d'intestazione per il True BASIC
	PW2TBsupport	Routine di supporto per il True BASIC
	PW2TBsupport*	Versione già compilata del file precedente
	README	Descrive l'uso dei sorgenti per il True BASIC
TDI		
	PW2TDI	Programma per la conversione dei file oggetto prodotti da PowerWindows in sorgenti TDI Modula 2
	PW2TDI header	File d'intestazione per il TDI Modula 2
	example.mod	Sorgente d'esempio in linguaggio TDI Modula 2
	PW2.def	File di supporto per il sistema TDI Modula 2
	PW2.mod	File di supporto per il sistema TDI Modula 2
	PW2.lnk	File di supporto per il sistema TDI Modula 2
	PW2.sym	File di supporto per il sistema TDI Modula 2
	README	Descrive l'uso dei sorgenti per il TDI Modula 2
Examples		
	8_simple_windows.pw	File oggetto d'esempio per l'uso delle finestre
	BM2example.pw	File oggetto d'esempio per il BenchMark Modula 2
	exemple.pw	File oggetto d'esempio per il C
	TDIexample.pw	File oggetto d'esempio per il TDI Modula 2
	ToolBox.pw	File oggetto d'esempio per l'uso di gadget multipli
	Wierd_menus.pw	File oggetto d'esempio per l'uso dei menu
	Wierd_screen.pw	File oggetto d'esempio per l'uso degli schermi

solo dei file oggetto creati da *PowerWindows* che possono essere usati dalle utility di conversione di ciascun linguaggio.

La directory più importante è sicuramente la directory C. Contiene due sorgenti creati da *PowerWindows*, uno per l'Aztec C e uno per il Lattice C, e un file README con importanti spiegazioni per l'uso di *PowerWindows* in relazione a entrambi i compilatori. Va comunque sottolineato che i sorgenti per questi compilatori vengono prodotti direttamente da *PowerWindows*, senza l'uso di programmi di conversione specifici, tramite l'opzione "Generate Source Code" presente nel menu Project. Scegliendo questa opzione vi verrà presentato un requester che vi permetterà di specificare il tipo di sorgente che preferite: Aztec C, Lattice C o Assembly. Per l'Assembly è inoltre presente un sorgente d'esempio nella directory Assembler.

Le altre directory contengono tutti i file di supporto per i vari linguaggi e un programma per la conversione dei file oggetto in sorgenti specifici. Per esempio, nella directory AB sono presenti quattro file con l'estensione ".bmap" che servono all'Amiga BASIC per la gestione delle librerie, un file con l'estensione ".header" per la produzione di file eseguibili e il programma PW2AB per la conversione.

Come funziona PowerWindows

Per mandare in esecuzione il programma *PowerWindows* basta semplicemente digitare il comando PW2.5 da una finestra CLI senza l'aggiunta di altri argomenti. I file oggetto d'esempio possono infatti essere caricati soltanto dopo aver attivato il programma. Durante l'attivazione viene invece caricato automaticamente il file "pw2- configuration" che dev'essere presente nella directory Devs. Questo file di configurazione può essere in seguito modificato e viene riscritto sul disco tramite l'opzione "Save Configuration" del menu Project. Ogni volta che un file oggetto viene salvato, assieme a esso viene salvata anche la configurazione attuale. In questo modo ogni volta che lo carichiamo la sua configurazione diventa la configurazione corrente per qualsiasi operazione successiva.

Non appena impostata la configurazione possiamo procedere alla creazione della nostra interfaccia grafica e quindi creare il file oggetto che verrà in seguito convertito in sorgente. Anche se per i linguaggi Aztec C, Lattice C e Assembly non serve salvare il file oggetto, in quanto si possono direttamente convertire da programma, è buona norma creare sempre tale file per ogni interfaccia che stiamo creando, nel caso volessimo apporre delle modifiche in una successiva sessione di lavoro.

I programmi di conversione

Per tutti gli altri linguaggi, il file oggetto è invece obbligatorio e viene richiesto dai programmi di conversione. Questi programmi permettono inoltre di conservare l'ambiente interattivo di *PowerWindows*, facilitando così le varie fasi di modifica, salvataggio e conversione dei file oggetto. Per una corretta conversione questi programmi necessitano anche di un file header (o d'intestazione) che contiene informazioni supplementari per il compilatore del linguaggio specifico e che viene agganciato al file oggetto prodotto da *PowerWindows* per ottenere un unico file sorgente completo e pronto per essere compilato. Tutti i programmi di conversione possono comunque utilizzare lo stesso file oggetto e quindi la scelta del linguaggio per il sorgente finale può essere rimandata alla fine della sessione di lavoro, quando cioè deciderete che la vostra interfaccia è completa e quindi pronta per essere convertita.

Generalmente parlando, la creazione di un'interfaccia grafica si può concepire come un unico processo (fase di edit) comprendente un insieme di varie fasi di lavoro creativo, ognuna delle quali rappresenta la costruzione di un particolare elemento dell'interfaccia stessa. L'interfaccia finale, così come apparirà nella versione definitiva del vostro programma, può essere stata generata mediante varie fasi di edit, separate fra loro da altre fasi di compilazione e correzione usate per verificare come tale interfaccia si integra col resto del vostro programma. La fase di edit comunque non deve necessariamente seguire uno schema rigido di sviluppo, in quanto come abbiamo detto è un processo completamente interattivo. Potete infatti salvare il file oggetto in qualunque momento e quindi potrete modificarlo ogni volta che lo riterrete opportuno. Inoltre potete passare tranquillamente da una fase di lavoro all'altra per definire le varie componenti della vostra interfaccia. Se per esempio state creando dei gadget e decidete improvvisamente che sarebbe più opportuno aggiungere anche una finestra, potete tranquillamente procedere alla creazione di tale finestra e riprendere poi in seguito la definizione dei gadget. Oppure potete salvare il lavoro svolto finora e cominciare la definizione dei nuovi elementi. Anche il caricamento dei file oggetto può essere effettuato in qualsiasi momento, tramite l'opzione "Load screen and/or windows" del menu Project, e quindi nella stessa fase di edit potrete usarne più di uno contemporaneamente.

Schermi, finestre, requester, menu e gadget

Osservando il menu Screen, troviamo innanzitutto l'opzione "Define type", selezionando la quale verrà visualizzato un requester per la definizione dello schermo che userete nella vostra interfaccia. Se per esempio desiderate che tutte le finestre del vostro programma vengano visualizzate su uno schermo di tipo custom piuttosto che su uno schermo Workbench, è necessario scegliere l'opzione "Custom screen" presente in questo requester. Si può inoltre definire la risoluzione, il titolo e i vari flag specifici di uno schermo. Una volta definito lo schermo, si può specificarne la palette di colori usando l'opzione "Screen palette". A questo punto il vostro schermo è pronto e tutte le finestre, menu, gadget e altri oggetti che verranno definiti in seguito vi verranno visualizzati.

Per la creazione delle finestre si può procedere in due modi: con l'opzione "Open a window" o con l'opzione "Grab a window" presenti nel menu Window. Usando l'opzione "Open a window" verrà visualizzato un requester per la completa definizione manuale della finestra, mentre con l'opzione "Grab a window" si accede a un altro menu tramite il quale si possono prelevare tutte le informazioni relative a una finestra già

presente, sia di *PowerWindows* sia di altri programmi.

Per modificare le caratteristiche della finestra così creata, si utilizza l'opzione "Edit window characteristics" sempre nel menu Window. Verrà così presentato un altro requester che permetterà di definire le dimensioni massime e minime, le dimensioni correnti, il titolo, la posizione corrente, i flag IDCMP di Intuition e i flag specifici della finestra. Inoltre si può determinare se rappresenta una finestra normale oppure un requester. Tutte queste scelte richiedono una certa familiarità con le caratteristiche proprie di Intuition. Nello stesso menu sono inoltre presenti delle opzioni per scrivere, muovere, modificare e cancellare del testo nella vostra finestra.

Tutte le precisazioni che sono state fatte per le finestre valgono anche per la creazione dei requester, in quanto un requester non è altro che un tipo particolare di finestra. Se scegliete il gadget "Requester" durante la creazione di una finestra potrete accedere soltanto alle opzioni relative; per esempio non potete specificare i flag IDCMP e altre caratteristiche che riguardano esclusivamente le finestre.

PowerWindows consente di disegnare i propri menu, ciascuno con i propri menuitem e subitem. Osservando il menu "Menus" notiamo tutte le opzioni che ci permettono di creare una lista di menu e di cambiare l'ordine di questa lista per ottenere quella che definisce nel migliore dei modi tutte le opzioni del nostro programma. Tramite l'opzione "Work on menuitems for..." si procede alla creazione dei menuitem per ciascun menu presente. Tutte le opzioni per la costruzione di questi menuitem sono presenti nell'apposito menu che comparirà nella parte alta dello schermo di *PowerWindows*. Com'è evidenziato nella Tavola 2, questo menu presenta delle opzioni supplementari rispetto a quello per la costruzione dei nostri menu. Potrete infatti definire il numero di colonne usate dai menuitem ed eventualmente associare a questi una immagine grafica che potrete caricare da disco, ovviamente in formato standard IFF ILBM. L'opzione "Work on subitem for..." vi permetterà inoltre di accedere a un altro menu per la costruzione dei subitem associati ai precedenti menuitem. Le opzioni per la costruzione dei subitem sono sostanzialmente identiche a quelle usate per i menuitem. In questo modo si possono ottenere delle complesse liste di menu contenenti ciascuna la propria lista di menuitems, che presentano a loro volta una ulteriore serie di subitems. Queste liste determinano la disposizione con cui i menu verranno in seguito visualizzati sullo schermo e permettono a *PowerWindows* di determinare gli effettivi collegamenti fra le varie strutture menu, menuitem e subitem che verranno in seguito definite nel file sorgente.

Nel menu Gadgets sono contenute le opzioni per la creazione di gadget proporzionali, di stringa e booleani (on/off) che potranno essere collegati alla finestra corrente. Inoltre *PowerWindows* fornisce due tipi di gadget supplementari, PopupGadget e DragGadget. Tramite queste opzioni possiamo aggiungere, muovere, duplicare, ridimensionare e cancellare dei gadget e ordinarli, secondo le nostre preferenze, all'interno di una lista specifica per ciascuna finestra. Dopo la creazione della lista possiamo passare alla definizione delle caratteristiche specifiche (bordi, testo e immagini) associate a ogni gadget. Facciamo comunque notare che i gadget racchiudono sempre delle zone rettangolari ben definite anche se sono associati a delle immagini o a dei bordi con contorni diversi.

PowerWindows dispone anche di un particolare tipo di gadget a forma circolare, chiamato knob, che ricorda una manopola di quelle usate nelle radio per la regolazione del volume. Le opzioni per la creazione dei knob sono comunque simili a quelle appena descritte per i gadget. Le caratteristiche proprie di questi knob vengono invece definite tramite l'opzione "Edit knob characteristics" presente nel menu Knobs. Tra queste abbiamo le coordinate X e Y del centro, il raggio del cerchio, la posizione corrente, l'ampiezza del movimento e tutti i flag per il suo controllo.

Le Preferences

La configurazione generale di *PowerWindows* può essere modificata in ogni momento durante la fase di edit attivando o disattivando le voci presenti nel menu Preferences. Se per esempio pensate che gli schermi di aiuto forniti automaticamente da *PowerWindows* rallentino il vostro lavoro, potete disabilitare la voce "OK Prompts", oppure se desiderate conoscere le coordinate del puntatore durante il posizionamento di un gadget, potete attivare la voce "Mouse coordinates". Ricordiamo che queste preferenze non solo possono essere salvate con l'opzione "Save configuration" del menu Project, ma vengono automaticamente inserite in ogni file oggetto che voi create mediante l'opzione "Save screen and/or window". Se durante la fase di edit caricate un file oggetto, la configurazione attuale verrà modificata dalle preferenze incluse in quel file e se queste non sono di vostro gradimento dovrete impostarle nuovamente tramite il menu Preferences. Le ultime due voci del menu Preferences non vengono usate durante la fase di edit ma sono importanti durante la procedura di conversione dei file oggetto in sorgente; se per esempio volete che *PowerWindows* inserisca dei commenti dettagliati all'interno del sorgente finale potete attivare la voce "Source code comments". Con la voce "Source code spacing (TAB)" viene invece definita la lunghezza delle tabulazioni usate nel sorgente per l'incolonnamento del testo e dei commenti.

Al lavoro!

Descriveremo ora come usare *PowerWindows* per creare velocemente un programma dimostrativo in C che permetta all'utente di fare delle scelte tramite menu.

La maggior parte delle operazioni di *PowerWindows* vengono svolte immediatamente sullo schermo; questo tipo di approccio viene generalmente chiamato WYSIWYG (What You See Is What You Get - ciò che vedi è ciò che ottieni). Se create un menu, con propri menuitem e subitem, potrete immediatamente vedere sullo schermo come esso risulterà nel vostro programma compilato. In ogni modo, anche se molte operazioni di *PowerWindows* sono graficamente concrete, la gestione interattiva degli eventi di menu è decisamente più astratta.

Come abbiamo visto, lo stesso *PowerWindows* ci fornisce delle opzioni per la creazione di tutti gli elementi di menu che serviranno al nostro programma.

Tavola 2: Sommario delle opzioni presenti in PowerWindows 2.5

Menu	Nome	Tasto	Descrizione
Project			
	Generate source code	Amiga-G	Genera i sorgenti nei linguaggi C e Assembly
	Save Configuration	Amiga-S	Salva la configurazione attuale nel file "DEFS pw2-configuration"
	Save screen and/or window		Salva su disco un file oggetto
	Load screen and/or window	Amiga-L	Carica da disco un file oggetto
	Kill everything		Cancella quanto è stato fatto nella corrente fase di edit
	Quit...	Amiga Q	Esce dal programma
Screens			
	Define screen type	Amiga-T	Crea un nuovo schermo
	Screen Palette	Amiga-P	Definisce la palette di colori per il nuovo schermo
Current window			
	Open new window	Amiga-N	Apre una nuova finestra
	Grab a window		Preleva una finestra dagli schermi degli altri programmi
	Redisplay	Amiga-R	Ridisegna la finestra corrente
	Clean up	Amiga-C	Pulisce la finestra
	Edit window characteristics		Definisce le caratteristiche della finestra
	Kill window	Amiga-K	Cancella la finestra
	Write window text	Amiga-W	Scrive del testo nella finestra
	Edit window text		Modifica il testo
	Move window text		Sposta il testo
	Delete window text		Cancella il testo
Menus			
	Append a menu	Amiga-M	Aggiunge un nuovo menu alla fine della lista dei menu
	Insert a menu before		Inserisce un nuovo menu
	Move a menu		Sposta il menu in un'altra posizione
	Edit a menu		Modifica un menu
	Work on menuitems for		Definisce il menuitem per uno specifico menu
	Delete a menu		Cancella un menu
Menuitems			
	Append a menuitem	Amiga A	Aggiunge un nuovo menuitem alla fine della lista dei menuitem
	Menuitems columns		Definisce il numero di colonne usate dal menuitem
	Insert a menuitem before .		Inserisce un nuovo menuitem
	Move a menuitem		Sposta un menuitem
	Edit a menuitem		Modifica un menuitem
	Work on images for		Definisce un'immagine associata al menuitem
	Work on subitems for		Definisce il subitem per uno specifico menuitem
	Delete a menuitem		Cancella un menuitem
Subitems			
	Append a subitem	Amiga-A	Aggiunge un nuovo subitem alla fine della lista dei subitem
	Subitems columns		Definisce il numero di colonne usate dal subitem
	Insert a subitem before		Inserisce un nuovo subitem
	Move a subitem		Sposta il subitem in un'altra posizione
	Edit a subitem		Modifica un subitem
	Work on images for		Definisce un immagine associata al subitem
	Delete a subitem		Cancella un subitem
Gadgets			
	Add a gadget	Amiga-A	Aggiunge un gadget alla lista dei gadget
	Move a gadget		Sposta il gadget in un altra posizione
	Clone a gadget		Duplica un gadget esistente
	Resize a gadget		Ridimensiona un gadget
	Define gadget type	Amiga-D	Definisce le caratteristiche del gadget
	Work on gadget borders		Definisce i bordi del gadget
	Work on gadget text		Definisce il testo del gadget
	Work on gadget images		Definisce l immagine del gadget
	Work on PopUpGadget menuitems		Definisce il menu a comparsa associato a un PopUpGadget
	Move gadget to start of list		Sposta il gadget all'inizio della lista dei gadget
	Choose gadget successor		Specifica la disposizione dei gadget nella lista
	Move gadget to end of list		Sposta il gadget alla fine della lista
	Delete a gadget		Cancella un gadget
Knobs			
	Add a knob		Aggiunge un knob alla lista dei knob
	Move a knob		Sposta il knob in un altra posizione
	Clone a knob		Duplica un knob esistente
	Resize a knob		Ridimensiona un knob
	Edit knob parameters		Modifica le caratteristiche di un knob
	Delete a knob		Cancella un knob
Preferences			
	OK prompts	Amiga-O	Attiva/disattiva i requester di conferma
	PW backdrop	Amiga-B	Mette lo schermo di PowerWindows sotto tutti gli altri schermi
	Auto redraw		Attiva/disattiva la visualizzazione automatica della finestra corrente
	Auto select		Attiva/disattiva la selezione automatica della finestra corrente
	Mouse coordinates	Amiga-X	Attiva/disattiva la visualizzazione automatica delle coordinate del mouse
	Gadget collision checking		Attiva/disattiva il controllo automatico delle collisioni tra gadget
	Image color remapping	Amiga-I	Attiva/disattiva la ridefinizione automatica dei colori di un immagine
	Trim extra bitplanes		Attiva/disattiva l eliminazione automatica dei bitplane eccedenti di un immagine
	Source code comments		Attiva/disattiva l inserimento automatico di commenti nel sorgente finale
	Source code spacing (TAB)		Definisce la spaziatura usata dalle tabulazioni del sorgente finale

A questo punto è importante notare che durante la fase di creazione dei menu è possibile associare una funzione in linguaggio C a ciascun menuitem e subitem. Scegliendo infatti l'opzione "Edit a menuitem" vi verrà presentato un requester che contiene tutte le caratteristiche del menuitem che state disegnando e tra queste è presente il gadget stringa Function nel quale potrete inserire il nome della funzione C simbolica. Supponiamo che nel vostro programma abbiate inserito una funzione chiamata "ClrScreen", che permette di cancellare lo schermo, e che vogliate attivarla tramite la selezione di un menuitem. Per ottenere ciò bisogna innanzitutto creare un menu e il relativo menuitem e in seguito, selezionando l'opzione "Edit a menuitem", inserire il nome "ClrScreen" nel gadget Function. Quando abbiamo terminato la definizione del nostro menu e delle funzioni a esso associate, possiamo passare alla fase di creazione del codice sorgente selezionando l'opzione "Generate source code" del menu Project. Nel requester che apparirà potremo scegliere sia il linguaggio che verrà usato per il sorgente, per esempio il Lattice C, sia gli elementi che desideriamo convertire; in questo caso attiveremo il gadget Menus. Inoltre, selezionando il gadget Event, verrà creata automaticamente una funzione in C chiamata HandleEvent che intercetta gli eventi di menu e passa il controllo alle funzioni appropriate. È proprio questa funzione che vi permetterà di unire in maniera semplice ed efficace la parte principale del vostro programma con il sorgente creato da *PowerWindows*.

Per evidenti ragioni di spazio, nel listato presentato nella pagina successiva viene descritta soltanto la parte principale del nostro programma dimostrativo, e cioè le funzioni necessarie per la gestione degli eventi di menu. Vengono comunque indicate le locazioni in cui potrete inserire sia delle sezioni scritte con il vostro editor di testi, per esempio quelle che riguardano l'apertura delle librerie, sia le sezioni di codice sorgente generate da *PowerWindows* che riguardano le descrizioni delle strutture usate. Studiate attentamente questi riferimenti per avere un'idea chiara di come costruire l'intero listato prima di procedere alla compilazione. Come potete vedere, il listato contiene un programma principale (main), la funzione HandleEvent e alcune funzioni fittizie per l'elaborazione degli eventi di menu.

Il programma principale usa la funzione SetMenuStrip per visualizzare il menu e procede con l'interpretazione dei messaggi che gli arrivano dalle routine interne di Intuition tramite la message port. Se questi messaggi sono di tipo CLOSEWINDOW, verrà impostato un flag che permetterà al programma di uscire al termine del ciclo di lettura dei messaggi stessi. Se invece sono di tipo MENUPICK, viene chiamata la funzione HandleEvent che a sua volta passa il controllo alle funzioni associate alla selezione dei menuitem.

Potete notare che il codice sorgente generato da *PowerWindows* per la funzione HandleEvent fa riferimento a tutte le strutture Menuitem del programma usando genericamente il nome "object". Questo puntatore può infatti essere associato a un qualsiasi oggetto di Intuition che possa generare degli eventi. Nel nostro caso la funzione HandleEvent prenderà come argomento il puntatore a una struttura MenuItem, ottenuto mediante una chiamata alla funzione ItemAddress, ma può essere estesa anche agli eventi provocati dai gadget, come vedremo in seguito. I nomi delle funzioni Menu1Item1Function e Menu1Item2Function sono quelli che devono essere specificati nei gadget Function associati a ciascun menuitem durante la preparazione dei menu tramite *PowerWindows*. Nel nostro semplice programma d'esempio queste due funzioni useranno il comando "printf" per visualizzare, nella finestra CLI corrente, un messaggio indicante la scelta effettuata dall'utente.

Il meccanismo appena descritto può essere inoltre usato per la gestione dei gadget. Se per esempio vogliamo attivare dei gadget up/down, che permettono cioè all'utente di far scorrere verticalmente il testo contenuto in una finestra, basterà specificare il nome delle funzioni appropriate nel requester che ci apparirà durante la definizione del tipo di gadget. In questo modo *PowerWindows* estenderà la definizione della funzione HandleEvent per la gestione supplementare di questi eventi.

Conclusione

Scoprirete presto che *PowerWindows* è un programma indispensabile per la creazione di programmi che fanno largo uso d'interfacce grafiche in stile Intuition, e che permette di risparmiare una grande quantità di tempo ed energia che si può invece dedicare alla valorizzazione della parte operativa del proprio programma. Il sistema usato da *PowerWindows* comporta infatti degli enormi vantaggi per tutti coloro che intendono programmare in modo serio l'Amiga in quanto usa tutte le risorse caratteristiche di Intuition.

Nel prossimo articolo

Nel prossimo articolo descriveremo il sistema di sviluppo in linguaggio C presentato dal Lattice Cross Compiler, versione 5.04.

Questo sistema permette ai programmatori dell'Amiga di scrivere e compilare i propri programmi in ambiente MS-DOS e OS/2 (80286 oppure 80386), usando le potenti caratteristiche di questi sistemi operativi, e successivamente trasportare il file eseguibile direttamente nell'Amiga per la correzione e l'esecuzione del programma stesso. Inoltre se il vostro Amiga è provvisto di una BridgeBoard o di una scheda Janus, potrete sfruttare nel modo più diretto i vantaggi offerti da questo sistema di sviluppo, ottenendo cioè le migliori prestazioni da ciascun ambiente operativo. Il Lattice C Cross Compiler viene distribuito su due dischi, uno in formato MS-DOS, oppure OS/2, e uno in formato Amiga. Ciascuno di questi dischi contiene inoltre tutti i più recenti file include dell'Amiga.

Per ulteriori informazioni contattare direttamente:

Inovatronics, Inc.
(PowerWindows 2.5, 89 dollari)
8499 Greenville Ave.
Suite 209B
Dallas, TX 75321
USA

Listato: ***PowerWindows_menu.c***

```
/* In questa sezione va aggiunto il sorgente
   generato da PowerWindows per la definizione
   delle strutture Screen, NewWindow, Window
   Menu, MenuItem, Gadget, Border,
   IntuiText eccetera. */


USHORT Quit_Flag = FALSE;

struct Window *Window1;
struct MenuItem *ItemAddress();
struct IntuiMessage *message;


/* ****************************************** */

main()
{
   USHORT code;
   ULONG class;

   /*    segue il vostro codice sorgente per
      l'apertura delle librerie, degli schermi,
      delle finestre eccetera ... */

SetMenuStrip(Window1, &Menu1);

   do
   {
      Wait(1L << Window1->UserPort->mp_SigBit);
      while((message = (struct IntuiMessage *)
        GetMsg(Window1->UserPort)) != NULL)
      {
         code  = message->Code;
         class = message->Class;
         ReplyMsg(message);
         if(class == CLOSEWINDOW) Quit_Flag =
           TRUE;
         if(class == MENUPICK)     HandleEvent(
            ItemAddress(&Menu1, (LONG) code));
      }
   } while(Quit_Flag == FALSE);

   ClearMenuStrip(Window1);
   if(Quit_Flag == TRUE) Quit();

   /*    segue il vostro codice sorgente
     per la chiusura delle librerie
     degli schermi, delle finestre
     eccetera ... */

}

/* ****************************************** */

/* [illegible]
   [illegible]
   [illegible] */
```

```
/* MenuItem1Function e MenuItem2Function
   rappresentano i nomi che sono stati assegnati
   ai gadget FUNCTION presenti nel requester
   "Edit a menuitem" di PowerWindows durante
   la costruzione dei menuitem. */

void HandleEvent(object)
APTR object;
{
   if(object == (APTR) &MenuItem1)
     { MenuItem1Function(object); return; }
   if(object == (APTR) &MenuItem2)
     { MenuItem2Function(object); return; }
}


/* ****************************************** */

/* Questa funzione va aggiunta al sorgente
   prodotto da PowerWindows. */

/* Definite quindi una funzione che chiude
   e disalloca tutte le risorse usate dal vostro
   programma prima di terminare la propria
   esecuzione. */

void Quit(object)
APTR object;
{
   Quit_Flag = TRUE;
}


/* ****************************************** */

/* Anche queste funzioni vanno aggiunte
   al sorgente prodotto da PowerWindows. */

/* In seguito potrete inserire delle specifiche
   funzioni che soddisfino le richieste
   dell'utente avvenute tramite la selezione
   dei menu. */

void MenuItem1Function(object)
APTR object;
{
   printf("E' stato selezionato il menuitem 1
     del menu.\n");
}

void MenuItem2Function(object)
APTR object;
{
   printf("E' stato selezionato il menuitem 2 del
     menu.\n");
}
```

■

ABBONATI!

A CASA TUA UN REGALO OGNI MESE

- Sicurezza di non perdere neanche un numero
- Prezzo bloccato per un anno
- Sconto del 15% sul prezzo di copertina

- Possibilità d'includere nell'abbonamento gli arretrati
- Comodità di ricevere la rivista direttamente a casa

Per chiunque sia interessato ai computer Commodore, *Commodore Gazette* è indispensabile. Nessuna rivista in Italia offre ai suoi lettori tanta qualità con recensioni hardware e software, listati, presentazioni esclusive, informazioni di ogni genere riguardanti C-64, C-128 e Amiga.

Un abbonamento a *Commodore Gazette* è il regalo più bello che possiate fare a voi stessi e agli altri... un regalo nuovo ogni mese. Ma non è finita! Risparmierete il 15% sul prezzo di copertina e potrete includere nel prezzo dell'abbonamento anche i numeri arretrati che mancano alla vostra raccolta. Resta inteso che per ogni arretrato scelto verrà spedito un numero di prossima uscita in meno. Per esempio, chi si abbona a 12 numeri a partire dal n. 1/89 e richiede 5 arretrati, riceverà 7 numeri del 1989 e 5 arretrati.

Ritagliare e spedire a: IHT Gruppo Editoriale – Via Monte Napoleone, 9 – 20121 Milano

Sì, desidero sottoscrivere un abbonamento a ☐ 12 numeri (L. 81.600) ☐ 24 numeri (L. 163.200) di COMMODORE GAZETTE usufruendo così dello sconto del 15% sul prezzo di copertina

Nome e cognome ....

Indirizzo .... Città .... CAP ....

Inizio abbonamento dal n. .... Desidero ricevere i seguenti numeri arretrati (specificare numero e anno) ....

☐ Allego assegno bancario o circolare ☐ Allego fotocopia della ricevuta del vaglia postale

Firma ....

Supra Corporation

SupraDrive™ 500XP

Complete Hard Disk/RAM System for the Amiga 500

SupraModem™ 2400

SupraRam™ 2000

SupraDrive™

SupraModem™ 2400zi

SupraModem™ 2400

SupraDrive

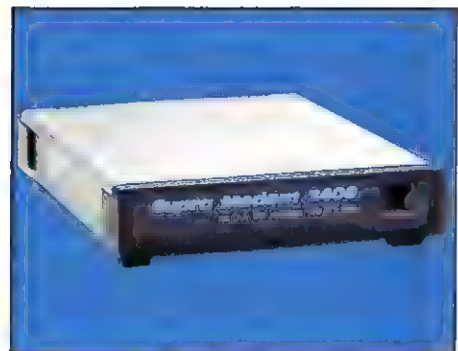
SupraModem™ 2400

Rende il vostro Amiga ancora migliore

SupraDrive™ Floppy

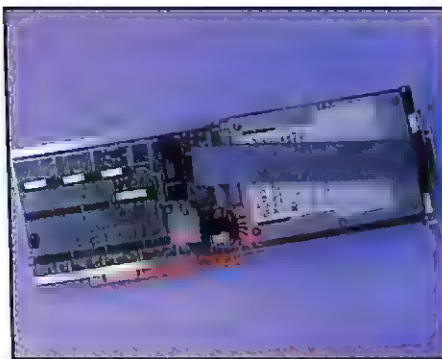
Silenzioso, affidabile floppy disk drive funzionante con tutti i computer Amiga. 1 MB di capienza non formattata. Porta passante per il collegamento di drive addizionali.

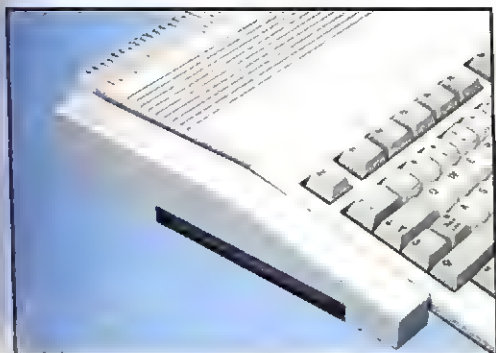
SupraDrive™ Removable

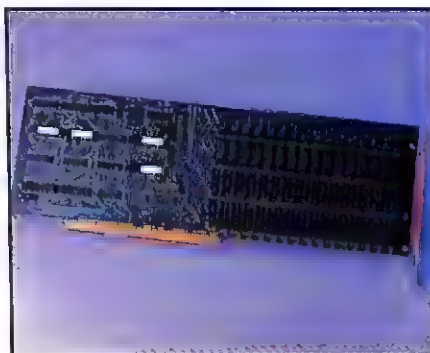
Drive Syquest a cartucce rimovibili per A500 o 2000. Ottimo per memorizzazione primaria e di back-up. Venduto con interfaccia SCSI o come drive aggiuntivo.

SupraModem™ 2400

100% Hayes-compatibile, modem a 300, 1200, 2400 baud per qualsiasi computer Compatibile con tutto il più popolare software per telecomunicazioni.

SupraDrive™ 500XP

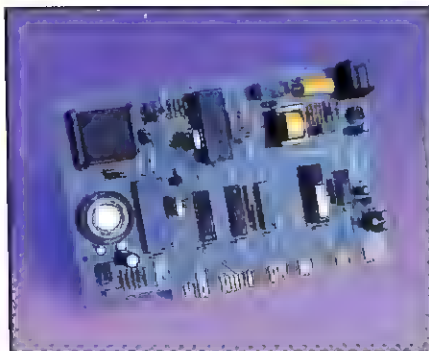
Hard disk drive alimentato dal computer, interfaccia SCSI & RAM aggiuntiva (espandibile da 1/2 a 8 MB) per A500. Include un bus Amiga passante, porta SCSI, software.

SupraDrive™ WordSync™

Scheda hard disk autoboot per l'Amiga 2000, di facile installazione. Usa un hard disk Quantum ad alte prestazioni e include porta SCSI & software di utilità.

SupraRAM™ 500RX

1/2, 1, 2, 4, o 8 MB di FAST RAM per l'Amiga 500. Facile da espandere. Zero wait state & hidden refresh. Bus Amiga passante.

SupraRAM™ 2000

2, 4, 6, & 8 MB di FAST RAM per l'A2000 & 3000. Facile da espandere. Zero wait state & hidden refresh. Scheda a quattro strati per una migliore affidabilità.

SupraModem™ 2400 Plus

SupraModem 2400 con correzione degli errori MNP5 & V.42bis e protocolli per la compressione dei dati. Consente trasmissioni fino a 9600 bps

SupraModem™ 2400zi

Modem interno su minischeda per Amiga 2000 & 3000. Si installa facilmente in qualsiasi slot Amiga bus. Supporta più modem in un solo computer.

Supra Corporation Distributori nel mondo

Canada
Beamscope
35 Ironside Crescent
Scarborough, Ontario M1X1G5
Tel: 416-291-0000
Fax: 416-291-5721

Finland
Westcom Systems Oy
Kirkkokatu 8, Fourth Floor
SF-48100 Kotka
Tel: 358-52-184-655
Fax: 358-52-184-007

Norway
Atlantis Distribution
Parallellen 12
8012 Bodoe
Tel: 47-816-3040
Fax: 47-816-3041

Spain
ABC Analog, S.A.
Santa Cruz de Marcenado, 31
28015 Madrid
Tel: 34-91248-8213
Fax: 34-1542-5059

Sweden
AlfaSoft AB
Magasingsgatan 9
S-216 13 Malmo
Tel: 46-40-164150
Fax: 46-40-163915

West Germany
European Software Distributors
Rodderweg 8
5040 Bruehl
Tel: 49-2223222-001
Fax: 49-2223222-003

Italia

I prodotti Supra sono disponibili nei migliori computershop e anche da:

Alex Computer
Torino - Tel. 011/7730184

Flopperia
Milano - Tel. 02/55188105

SUPRA CORPORATION
1133 Commercial Way - Albany, OR 97321
USA - TEL. 001/503/9679075
FAX: 001/503/9269370

IL NUMERO DI DISPOSITIVO DEL DISK DRIVE

Immaginate di modificare i numeri di dispositivo dei disk drive al semplice tocco di due tasti!

di Mark Jordan

Il numero di dispositivo del disk drive è qualcosa che la maggior parte degli utenti cambiano una volta sola: quando acquistano un secondo drive. E questo semplicemente perché ogni drive collegato al computer ha bisogno di un numero differente. Altrimenti l'elaboratore sarebbe confuso.

I disk drive della Commodore offrono due metodi per cambiare il numero di dispositivo: via software e via hardware. La maggior parte degli utenti si avvalgono del metodo hardware, ma ci sono buone ragioni per imparare a effettuare la modifica via software: per aggirare l'inflessibile impostazione di numero di drive del C-128D, per fare in modo che programmi che funzionano solo sul dispositivo numero 8 funzionino sul secondo drive (specialmente il 1581) senza dover agire su micro-interruttori e in generale per permettere un uso più efficiente di due disk drive.

Personalmente, ho dovuto confrontarmi con questo problema quando il disk drive interno del mio 128D ha smesso di funzionare correttamente. Questo ha reso inutilizzabile la maggior parte del mio software, dal momento che l'impostazione standard del numero di dispositivo di questo drive è 8. Per risolvere il problema, non potevo neanche cambiare il numero del drive esterno in 8, perché il drive interno del 128D continuava a essere 8. Così, capii che l'unica soluzione possibile era via software.

Prima di spiegare come fare, parliamo un attimo dei numeri di dispositivo. Ogni periferica ha bisogno di un suo numero. Tutte le volte che si apre un file, è necessario specificare questo numero, in modo che il 128 sappia dove convogliare i dati. La Commodore permette che siano collegati e indirizzati fino a 31 dispositivi (con numero 0-30), ma i numeri 0-7 sono riservati per altre periferiche. Il che lascia i numeri 8-30 per i disk drive. Mi sembra davvero abbastanza; non posso immaginare un utente con 23 disk drive collegati a un solo computer.

La modifica software è descritta nell'Appendice A del manuale del disk drive 1571 e può essere effettuata con una riga d'istruzioni:

```
OPEN15,8,15,"U0>"+CHR$(9):CLOSE 15
```

La linea è semplice, ma criptica. Usiamo OPEN per accedere alla periferica di cui abbiamo bisogno. Il primo numero è quello di canale, che può essere un numero qualsiasi tra 1 e 255. Dal momento che il terzo numero dev'essere 15, il primo numero può essere anch'esso 15, rendendolo così più semplice da ricordare.

L'8 tra i due 15 indica che stiamo "parlando" al dispositivo 8. Abbiamo assunto che si voglia cambiare il dispositivo 8 in un altro numero. Se si fosse voluto cambiare un numero di dispositivo diverso, per esempio 9, si sarebbe dovuto indicare 9.

Il secondo 15 è l'indirizzo secondario, ed è un numero importante, perché 15 è riservato all'invio di messaggi all'unità disk drive fisica.

Il comando "U0>" dice al drive di preparasi per una nuova direzione. L'istruzione +CHR$(9) dice al drive in quale numero cambiare il dispositivo 8. CLOSE 15 chiude semplicemente il canale.

La riga d'istruzioni funziona correttamente se tutto quello che si deve fare è cambiare il dispositivo 8 in 9, ma quasi ogni volta che si cambia un numero di dispositivo se ne cambia un altro. Quello che invece si vuole fare nella maggior parte dei casi è semplicemente scambiare i numeri di dispositivo di due disk drive; il che vuol dire che l'8 diventa 9 e viceversa.

Il problema nel cambiare due numeri di dispositivi è che possono finire per avere temporaneamente lo stesso numero, e a quel punto non è più possibile terminare la transazione, perché risponderanno entrambi i disk drive. Date uno sguardo:

```
OPEN15,8,15,"U0>"+CHR$(9)
```

In questo modo non succede nulla di buono: come questa linea viene eseguita, si ottengono due dispositivi 9. Se a questo punto si tenta di modificare l'originale 9 in 8, rispondono entrambi i drive.

Il trucco consiste nel cambiare prima uno dei due in un terzo numero, come segue:

```
OPEN15,8,15,"U0>"+CHR$(20):CLOSE 15
```

Il che va bene, in quanto sicuramente non si ha un altro drive con numero di dispositivo 20. Ora cambiamo il drive 8 in 9:

```
OPEN15,9,15,"U0>"+CHR$(8):CLOSE 15
```

Ora il secondo drive è 8 e il primo è 20. Finiamo il tutto cambiando il primo drive in 9.

```
OPEN15,20,15,"U0>"+CHR$(9):CLOSE 15
```

Ora è tutto invertito, e tutte le volte che si usa un comando sul drive di default (8), come Directory o Run, quello che prima era il secondo drive si metterà a girare, mentre il primo rimarrà in silenzio. E questo è il modo con il quale potete continuare a usare un C-128D con il drive interno rotto con un disk drive esterno.

Come ho già detto, ci sono altri vantaggi con questi micro-interrutto-

ri software. Dal momento che molti programmi su dischi da 5,25" funzionano solo con il dispositivo 8, trasferirli su un disco del 1581 non porta a nulla di buono se non ci si occupa costantemente di girare il drive e di agire sui suoi micro-interruttori. È più pratico utilizzare un micro-interruttore software: numerate le righe sopra esposte in ordine crescente in modo da formare un programmino, salvatelo come file sul 1581 e mandatelo in esecuzione prima di caricare il programma desiderato.

I micro-interruttori software possono anche consentire un uso più efficiente del proprio computer. Nelle righe che seguono è presentato un piccolo programma in linguaggio macchina, *DEVICE.SWITCH*, che scambia i dispositivi 8 e 9 alla pressione dei tasti shift/restore. In fase di programmazione questo può far risparmiare tempo, perché si può usufruire di due dischi e si possono scambiare dati con entrambi senza dover digitare i comandi extra U richiesti dal Basic.

Per utilizzare la routine, prima digitatela e salvatela, poi inserite un disco e mandatela in esecuzione. Ne verrà salvata su disco una versione in linguaggio macchina dal nome di *DEVICE.SWITCH*. Ogni volta che vorrete eseguirla, effettuate il BLOAD di *DEVICE.SWITCH*, digitate SYS 2912 e premete Return. Tenete presente che qualche programma eseguito successivamente può scriverci sopra e cancellare il nostro programmino dalla memoria.

Sono sicuro che la modifica dei numeri di dispositivo può avere molti altri usi. Li lascio scoprire a voi.

Istruzioni: digitare il listato, salvarlo su disco e impartire il run.
Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe. Per ulteriori informazioni vedere "Come digitare i listati" a pagina 94.

```
10 FORT=0TO120:READA$:POKE2816+T,DEC
(A$):NEXT
20 BSAVE"DEVICE SWITCH",B0,P2816 TO
 P2937
30 REM LE RIGHE 30-100 DIMOSTRANO L
EFFETTO
40 SYS2912
50 KEY1,"DI{SHFT R}"+CHR$(13)
60 KEY[illegible],"DI{SHFT R}[illegible]"+CHR$(13)
70 PRINTCHR$(14)"{SHFT T}{SHFT R}{SHFT P}REM
I {SHFT S} {SHFT H} {SHFT I} {SHFT F} {SH
FT T} {SHFT R}UN/{SHFT S}TOP PER SCAMBI
ARE I DRIVE"
80 PRINT"{SHFT F}PREMI {SHFT F}1 PER VED
ERE LA DIRECTORY DEL DISPOSITIVO 8"
90 PRINT"O {SHFT F}3 PER IL DISPOSITIV
O 9"
99
1000 DATA 4C,10,0B,55,30,3E,0[illegible],55,30
 3E,0[illegible],55,30,3E,0[illegible],EA
1010 DATA A9,0F,A2,09,A0,0F,20,BA,FF
,A9,04,A2,07,A0,0B,20
1020 DATA BD,FF,20,C0,FF,A9,0F,20,C3
 FF,A9,0F,A2,08,A2,0F
1030 DATA 20,BA,FF,A[illegible],34,[illegible],07,A0,0B
 20,BD,FF,20,C0,FF,A9
1040 DATA 0F,20,C3,FF,A9,0F,A2,0[illegible],A0
,0F,20,BA,FF,A9,04,A2
1050 DATA 0E,A0,0B,20,BD,FF,20,C0,FF
 A9,0F,20,C3,FF,20,E[illegible]
1060 DATA 78,A9,[illegible]C,8D,[illegible],03,A9,0B,8D
 1[illegible],03,[illegible],60,AE,D[illegible],[illegible]C
1070 DATA [illegible]1,[illegible],0[illegible],[illegible]0,10,0B,4[illegible],42,F[illegible]
```

COMMODORE E DIDATTICA

ESPERIENZE DIDATTICHE A CONFRONTO

LA FORMAZIONE DEI DOCENTI

L'attivazione, con l'Ordinanza Ministeriale n. 282 del 10 agosto 1989, della figura dell'Operatore Tecnologico nella Scuola dell'obbligo, nonché la recente Riforma degli Ordinamenti della Scuola Elementare, ripropongono con forza almeno pari alla sua indefettibile urgenza e necessità, il problema dell'aggiornamento e della formazione degli insegnanti.

È un problema assai delicato poiché investe e pone in essere variabili sovente in conflitto tra loro e, per questa ragione, difficilmente gestibili. La natura di queste variabili può essere: a) numerica: sono molti i docenti da formare o, se si vuole, da "alfabetizzare"; b) tipologica: diversi e non pochi sono gli insegnanti, le materie che insegnano, il grado di approfondimento del loro insegnamento; c) organizzativa: riguardante cioè il come organizzare il corso, il come e che cosa presentare dell'Informatica tenendo presente le variabili di cui al precedente punto; d) temporale: ossia il tempo, quantificato in ore, che l'aggiornatore ha a disposizione per svolgere il corso di formazione (questo tempo può includere: 1) lezione frontale dell'esperto; 2) eventuali lavori di gruppo degli aggiornandi; 3) eventuali esperienze in classe sugli argomenti le metodologie e le tecniche didattiche svolte durante il corso; 4) incontro con aggiornatore per valutare e discutere la parte del corso); e) logistica: sembra d'importanza relativa ma, in ogni caso, è bene non sottovalutarla: è intuibile che il luogo (o i luoghi) dove verrà vissuto un certo evento (nel nostro caso il corso di formazione e/o aggiornamento in Informatica) dev'essere comodo e funzionale all'evento stesso. Sarebbe antipatico se, dopo aver pensato a tutto, ci si accorgesse che non ci sono prese a sufficienza per i computer da installare (anch'essi in numero sufficiente) o che i KWatt forniti dall'ENEL non sono sufficienti se tutte le macchine e le periferiche collegate sono accese. Per quanto rapidamente (ma non frettolosamente), è meglio assicurarsi, in fase di organizzazione, che cose del genere non possano accadere e scegliere quindi l'edificio scolastico che, da questo punto di vista, offra maggiori garanzie; f) economica: non solo nel senso di avere risorse per pagare gli esperti, gli eventuali materiali di cancelleria (all'occorrenza: fogli per cartelloni murali, lucidi e pennarelli per la lavagna luminosa, pennarelli normali, carta...) o altro ma, anche, per dotare le scuole dei docenti corsisti di hardware e software necessari per poter mettere in pratica con le proprie classi quanto appreso. È una questione che non ci sembra esagerato definire d'importanza capitale. È facilmente comprensibile, infatti, che, quand'anche ci siano buona volontà, interesse e impegno da parte degli aggiornandi, l'efficacia teorica, pratica e, diciamolo pure, innovativa, del corso, verrà vanificata se non si potrà poi disporre di computer e di software nella propria scuola: sarebbe un po' come prendere la patente e, dopo, non avere l'automobile; a poco a poco si disimparerebbero tutti i consigli e gli accorgimenti imparati durante le ore di guida; g) psicologica: è la variabile che meno si tiene in considerazione ed è invece fondamentale per l'uso reale che il docente corsista farà del computer a scuola: non ci soffermeremo qui a illustrare nuovamente gli atteggiamenti, le idiosincrasie degli insegnanti nei confronti dell'elaboratore già trattati in questa rubrica.

Questa rubrica si basa su testimonianze di alunni ed insegnanti, che si interessano alla didattica del computer. I contributi editoriali sono grandemente aprezzati. Inviate eventuali materiali (articoli, foto, disegni, descrizioni di esperienze...) a:

COMMODORE GAZETTE
Comodore e didattica
Via Monte Napoleone, 9
20121 Milano

Formatori e formati

Tutte queste variabili contribui-

scono alla buona riuscita e, quindi, all'efficacia di un corso di formazione e/o aggiornamento in Informatica destinato ai docenti, sempreché, naturalmente, sia stata, per così dire, "assegnata" prioritariamente la variabile senza la quale sicuramente il corso non si può realizzare. Quella riguardante, cioè, il chi terrà il corso agli insegnanti. Il Piano Nazionale per l'introduzione dell'Informatica nelle Scuole Secondarie Superiori prevedeva (e tuttora prevede) a questo proposito due momenti: a) una formazione degli aggiornatori (per ragioni eufoniche non diciamo "formazione dei formatori") ai quali sarebbe stata affidata la b) formazione vera e propria degl'insegnanti; in prima istanza quelli di Matematica e Fisica "che già possiedono conoscenze informatiche" ("Il Piano Nazionale per l'introduzione dell'Informatica nelle Scuole Secondarie Superiori", 1985, p.61), coinvolgendo successivamente e con più gradualità "anche gli insegnanti delle altre discipline perché introducano l'uso dell'elaboratore a supporto della didattica" (ibidem). Dall'86 così si è proceduto e, malgrado le statistiche dimostrino come, agendo in questo modo, sia stato possibile formare una quantità ragguardevole di insegnanti, pare sia l'aspetto qualitativo (ossia il "dopo" a scuola) a lasciar a desiderare, non tanto per la scarsa competenza degli esperti quanto per i pressoché cronici disguidi organizzativi che si verificano sovente nella quotidianità scolastica cui si assommano i pregiudizi, il senso d'estraneità, lo scetticismo di alcuni docenti (i computer son stati ordinati ma ci sono ritardi nella consegna: chissà quando arriveranno, oppure ci sono ma si ripresenta lo stesso problema per il software; l'uso del computer spetta all' assistente di laboratorio, col metodo con cui insegno io, non serve). Vi è, però, una ragione più intrinseca dell'attrito tra impiego dell'elaboratore come sussidio didattico e insegnamento. In altre parole, mal si conciliano la flessibilità e la versatilità del computer con la linearità – abbastanza rigida – non soltanto del programma da svolgere ma, anche, del modo d'insegnare ossia del trasmettere la/le conoscenza/e: si pensi alle possibilità di presentazione e organizzazione di queste offerte dai cosiddetti ipertesti. Queste difficoltà possono davvero gettare ombre di crisi nelle pur buone intenzioni del Piano e fornire occasioni di critica a chi all'efficacia didattica del computer ha sempre creduto poco (per non dire mai).

La formazione degli O.T.

Secondo la C.M. n. 283 del 10 agosto 1989, la strategia di formazione degli Operatori Tecnologici (O.T.) per la Scuola dell'obbligo si articola in quattro punti: a) la costituzione di un'équipe di direzione avente principalmente il compito organizzativo di coordinare, controllare e favorire il collegamento tra i vari corsi e le sperimentazioni che li accompagnano, poiché, dato l'alto numero di personale da formare, si è subito visto quanto sia preferibile "limitare al minimo i pur fondamentali momenti corsuali unitari e accentrati" e puntare il più possibile su "strutture di progetto disseminate nel territorio"; oltre a questa équipe, poi, se ne dovrà costituire un'altra avente il preciso scopo b) "di produrre organici pacchetti formativi che possano essere direttamente applicati nei singoli corsi da operatori di medio livello (per esempio docenti di scuola secondaria)"; c) la predisposizione dei corsi in sedi scolastiche idonee, già fornite, cioè, delle attrezzature necessarie; d) l'affidamento della conduzione dei corsi a una terza équipe composta da non più di tre insegnanti "specializzati rispettivamente in problemi di ordine culturale-didattico (programmazione), in problemi di applicazione dei linguaggi audiovisuali e in problemi di applicazione operativa dell'informatica".

A leggerle così, sulla carta, queste direttive non sembrano superficiali. Fatto sta, però, che, dalle informazioni in nostro possesso, nell'anno scolastico 1989-90, esse sulla carta son rimaste: per coloro che erano riusciti a ottenere dal Provveditore la nomina di Operatore Tecnologico si son organizzati tutt'al più stage intensivi di qualche giorno. Tenuto conto, anche, della limitazione di base (di per sé piuttosto sconcertante) di questa nuova figura professionale (il fatto che la legge 426/88 – art. 5, I comma, legge con la quale viene costituita – ne decreti il reperimento "nei limiti del 20 per cento del personale soprannumerario"), questa mancata realizzazione di una formazione generalizzata degli O.T. sul territorio nazionale, ha fatto ipotizzare agli interessati (già O.T. oppure aspiranti tali) una sotterranea intenzione da parte del Ministero d'insabbiare e far gradatamente morire questa figura professionale, nonostante nella C.M. n. 283 sia testualmente scritto che "se attualmente è collegata alla situazione contingente di soprannumero in alcuni settori di scuola, in prospettiva essa non deve essere vincolata a tale situazione". Speriamo! Anche se non si riesce a vedere che attributi di concretezza possa avere questa prospettiva se, per legge (difficile da cambiare o da abrogare), ci si dovrà sempre basare sulla percentuale di docenti soprannumerari. Il che significa che se nella provincia non vi è una situazione di soprannumero in tale provincia non ci saranno Operatori Tecnologici. È un vero peccato, perché, riflettendoci sopra e col passare del tempo, ci si convince sempre di più dell'utilità (se non dell'indispensabilità) di una simile figura. Indirettamente, lo sta a dimostrare una sperimentazione sulla formazione dei docenti condotta in Inghilterra.

Esperienze da considerare

Nella sua relazione al Convegno "Strumenti e itinerari per il trattamento dell'informazione nella Scuola" (Convegno organizzato all'interno della manifestazione "Scuola 2000 computer sul banco", svoltasi dall'1° al 4 marzo 1989 a Milano) Richard Noss, dell'Institute of Education dell'Università di Londra, riferisce la sua esperienza come aggiornatore degli insegnanti che hanno partecipato al "Microworlds Project" riguardante il LOGO e la Matematica. In essa vengono confermate molte delle difficoltà già note (la diffidenza, almeno iniziale, di alcuni docenti all'uso del computer, talvolta, la scarsità delle macchine installate nelle scuole...). Ma ciò che è più interessante, è l'atteggiamento che, come aggiornatore, decise di assumere nei confronti dei docenti. "Abbiamo tentato", scrive Noss, "di porre i nostri studenti

insegnanti nel ruolo di aggiornatori, incoraggiandoli a diffondere la loro conoscenza e i loro materiali nei loro dipartimenti, nelle loro scuole e oltre. Nel fare questo eravamo perfettamente consci di ribaltare una delle idee cardine ormai accettate (almeno nel Regno Unito), che il cambiamento abbia luogo per mezzo di una "cascata" dall'alto; al contrario il nostro approccio era basato su un accostarsi dal basso verso l'alto che poggiava a sua volta sulla convinzione che praticanti informati e impegnati sono essi stessi nella posizione più forte per realizzare il cambiamento. La nostra funzione di esperti la vediamo molto come coloro che forniscono una sorta d'influsso catalitico in grado di diffondere il cambiamento dal basso non per una qualche particolare ragione di principio, ma piuttosto per una questione di adattamento; l'esperienza del Regno Unito ha ampiamente dimostrato (e continua a dimostrare) che un approccio dall'alto è inefficiente e inefficace".

Nelle sue affermazioni, Noss esprime tre idee meritevoli, a nostro parere, di una riflessione. La prima, l'idea di aggiornare, disseminando così il cambiamento, procedendo "a cascata", fatta propria anche dal nostro Piano Nazionale, si rivela poco fruttuosa contrariamente alla seconda di adattarsi più convenientemente e opportunamente alla realtà vissuta degli aggiornandi, partendo, perciò, dalle loro condizioni culturali e ambientali senza dare nulla per scontato. A questo proposito, è da rilevare che ben di rado, per non dire mai (almeno mai ne abbiamo sentito parlare), nel pianificare questi (e altri, di altre discipline) corsi di formazione/aggiornamento, ci si è data la pena di sottoporre i futuri corsisti a un test con lo scopo di accertare ciò che eventualmente già sapessero del computer come macchina e del suo impiego didattico, in modo, anche, da rendersi conto dell'opinione, delle convinzioni, dei pregiudizi che gli aggiornandi si fossero fatti sull'argomento da affrontare. I risultati di un simile test indurrebbero a constatare, probabilmente, che la prima cosa (piuttosto impegnativa) da fare sarebbe quella di cercare di sradicare dalle loro menti *les idées recues* nei confronti dell'elaboratore: non tanto quelle dei professori di Matematica e Fisica "che già possiedono conoscenze informatiche", primi destinatari del Piano, quanto quelle di coloro che insegnano altre materie e che non necessariamente frequentano (o frequenteranno) i corsi del Piano, ma anche altri, organizzati e gestiti da altri enti od organismi; bisogna anche ipotizzare che non è assolutamente certo che chi frequenta uno di questi ultimi corsi (in genere facoltativi), per il semplice fatto che vi si iscrive spontaneamente è di per sé motivato e privo di pregiudizi. Il test d'entrata, inoltre, implica una progettazione del corso a più livelli: poniamo, di base, intermedio e avanzato. La rilevazione del test, infatti, potrebbe anche indicare che tra i corsisti vi è chi, per interesse personale, ha già qualche nozione d'Informatica e pratica di tastiera. In casi come questo, sarebbe avvilente, mortificante e, soprattutto, demotivante per il futuro aggiornando vedersi assegnato a un corso di primo livello; naturalmente, sarebbe anche un evidente sbaglio sotto il profilo psicopedagogico e didattico. La terza idea che scaturisce dall'esperienza inglese è quella riguardante il tentativo di trasformare il corsista stesso in teacher-educator, in aggiornatore dei propri colleghi nonché divulgatore dei materiali prodotti (Unità Didattiche, uso di programmi, procedure LOGO...) durante il corso. Molto realistica, a ben pensarci, è l'idea implicita secondo la quale "praticanti informati e impegnati sono essi stessi nella posizione più forte per realizzare il cambiamento" se non altro perché questi colleghi, sicuramente più dell'esperto venuto dall'Università, conoscono molto bene il loro ambiente di lavoro, l'atmosfera che lo caratterizza, le possibilità e le risorse, anche finanziarie, esistenti; e, così, forti di questa consapevolezza, possono essere più agevolmente in grado di trovare la strategia d'approccio più efficace e convincente per e con i loro colleghi. Oltre a ciò, al contrario dell'esperto che, spesso, dopo aver tenuto il corso, non si rivede più, il collega "informato e impegnato" può dare alla scuola un'assistenza continua per tutto l'A.S., attuale e, salvo trasferimenti, per quelli a venire. È un'idea che avevamo ventilato tre anni fa e che in qualche modo s'avvicina alla figura dell'Operatore Tecnologico. Bisogna comunque tener presente che le asserzioni di Noss fanno riferimento all'ambiente britannico e che il campo della sua sperimentazione è ristretto alla Matematica e ai micromondi LOGO. Il che ci fa pensare che, nella realtà scolastica italiana, il principio del teacher-educator è realizzabile più che altro nella Scuola dell'obbligo; già meno nella Secondaria Superiore (ma ciò non vuol dire un'impossibilità totale), dove la distinzione tra discipline è più marcata, considerato, anche, il maggior grado di approfondimento delle medesime in questo tipo di scuola.

Il buonsenso suggerirebbe che, a parte quelli del Piano Nazionale, se vengono organizzati anche altri corsi da altri enti e organismi (e questo può valere anche per la Secondaria Inferiore), l'esperto sia un docente della stessa materia insegnata dai corsisti. In altre parole, un insegnante di Matematica o anche d'Informatica (scartando a priori la possibilità che l'esperto sia un rappresentante di una ditta di computer o, anche, una persona di quelle scuole private che offrono corsi d'Informatica sovente per impiegati aziendali) ben poco avrà da dire ai colleghi di Lettere sull'uso didattico dell'elaboratore per la Linguistica e le Materie Letterarie: potrà tutt'al più illustrare come si usa tecnicamente un word processor per cui sarebbe sufficiente un buon manuale in italiano, non certo l'impiego e il valore didattici del word processor nella composizione di un testo o nella creazione di esercizi (approntando, magari, esempi pratici immediatamente fruibili in classe). Lo stesso discorso lo si potrebbe fare per l'uso del database e del foglio elettronico in Storia o del CAD nel disegno tecnico o della simulazione in Scienze, o, persino, in Educazione Civica. Di qui si può ben dedurre che mai viene la macchina prima della Didattica: è sempre il contrario. Ne consegue che è il professore di quella data materia che – conoscendo le versatili possibilità offertegli dall'elaboratore e dal software – intuisce la maniera in cui potrebbe usare quello specifico pacchetto o programma per i propri scopi didattici. Vista da

questa angolazione, la capacità manuale di usare un computer e la conoscenza dei principi tecnici di funzionamento (bit & byte, bus dei dati, bus degli indirizzi, microprocessore e così via) assumono un'importanza relativa e decisamente secondaria: per un insegnante non son più importanti che per un automobilista aver una perfetta conoscenza del motore e delle sue parti.

Concludendo

Con queste riflessioni si è cercato di mettere a fuoco un problema complesso per la cui soluzione non sembrano esistere facili ricette. Come si sarà intuito, non sono coinvolte componenti soltanto numeriche e culturali ma anche finanziarie, contenutistiche, psicopedagogiche e didattiche, che, idealmente, dovrebbero interagire in armonica sinergia. L'esperienza britannica citata ci ha dato suggerimenti preziosi, che, a nostro avviso varrebbe la pena mettere in pratica: potrebbero dare, forse, se non soluzioni definitive, almeno un migliorato rapporto tra insegnamento e uso didattico dell'elaboratore.

(di Stefano Franzato)

RIFERIMENTI BIBLIOGRAFICI. 1985 "Il Piano Nazionale per l'introduzione dell'Informatica nelle Scuole Secondarie Superiori" in Informatica nella scuola, "Studi e documenti n. 32" degli Annali della Pubblica Istruzione, Firenze, Le Monnier, pp.45-72. FRANZATO, STEFANO: 1987 "Una brutta malattia: l'ansia informatica", Commodore Gazette, II, n. 6, Milano, IHT Gruppo Editoriale, pp.118-120. 1990 a) "La paura di evolversi", Commodore Gazette, V, n. 2, Milano, IHT Gruppo Editoriale, pp.120-1; b) "Superare il tecnocentrismo: riflessioni e suggerimenti", NTE - Compuscuola, VI, n. 46, Milano Gruppo Editoriale Jackson, pp.26-8. GIATTI, PAOLO: 1988 "Vicenza e dintorni", Informatica, Telematica e Scuola, nn. 11/12, Milano McGraw Hill, pp.6-7. MAURI, GIANCARLO: 1988 "Ipertesti, mille percorsi del sapere", La tartaruga - Scuola & Informatica, I, n. 2, Roma, Armando, pp.16-9. NOSS, RICHARD: 1990 "Microworlds: the Need for Teacher Education", NTE - Compuscuola, VI, n. 45, Milano, Gruppo Editoriale Jackson, pp.32-40.

ERRATA CORRIGE

Stefano Franzato, autore della rubrica "Computer e Didattica", in merito all'articolo apparso sul numero 4/90 di *Commodore Gazette*, ci segnala che a pagina 88, terza colonna in alto, la frase corretta è da intendersi invece la seguente: "Ci piace questo inciso: "di regola", poiché riconosce ufficialmente ciò che abbiamo sempre saputo e scritto – anche su questa rubrica – e che non perdiamo occasione di ribadire. Certo, si tratta di una realtà...".

COMPUTER NEWS

NOVITÀ HARD E SOFTWARE DALL'ITALIA E DAL MONDO

ITALIA

ITALY AMOS USERS CLUB

Un gruppo di programmatori italiani, impegnati nello sviluppo di un ambizioso progetto nel campo della didattica e che hanno adottato con entusiasmo il linguaggio AMOS per il loro lavoro, hanno fondato un club italiano dedicato alla diffusione di notizie, trucchi di programmazione, software PD e di tutto quanto concerne lo sviluppo di software con AMOS. Chiunque volesse ulteriori informazioni, oppure volesse diffondere sul mercato di PD realizzazioni personali, è invitato a inviare le singole richieste e/o i propri lavori (accompagnati dalle opportune note di documentazione possibilmente anche in lingua inglese), al seguente indirizzo:

Avelino De Sabbata
Via G. Carducci, 3
33050 Pozzuolo del Friuli (UD)
(Tel. 0432/560426)

Nell'attesa che venga istituito un BBS dedicato prevalentemente al club, gli utenti Fido possono inviare i propri messaggi utilizzando l'area echo nazionale IAMIGA, oppure utilizzando il MATRIX all'indirizzo FidoNet riportato di seguito:

Avelino De Sabbata 2:333/26.0

TOTOCALCIO, TOTIP ED ENALOTTO

La Progetto Software ha realizzato un pacchetto per lo sviluppo e l'elaborazione di sistemi di Totocalcio, Totip e Enalotto per Amiga.

Il nome del programma è *Sisthema* e porta sull'Amiga le più avanzate tecniche di elaborazione dei sistemi a pronostico, tra cui i sei tipi di condizionamento (totale segni globali, consecutività differenziata, interruzioni, formule derivate per inclusione ed esclusione, correzione d'errori a livelli ibridi, colonne filtro) e i due esclusivi metodi di riduzione per sistemi integrali e condizionati. La versione Plus ha anche la possibilità di stampare le colonne direttamente sulla shcedina. Il prezzo di listino è di 70.000 lire per la versione base, e 159.000 lire per la versione Plus (prezzi Iva esclusa).

Progetto Software
Via Rodi, 39
10095 Grugliasco (TO)
(Tel. 011/700358)

COMPUTER LAB NEWS

Il 27 settembre 1990 il centro di assistenza autorizzato Commodore per Milano e Lombardia, Computer Lab, ha promosso un incontro con la stampa specializzata per presentare il suo nuovo laboratorio in Via Cadore 6, a Milano. La Computer Lab propone ai suoi clienti una serie di servizi interessanti: verifica del difetto dichiarato al momento dell'accettazione in presenza del cliente; collaudo al momento del ritiro; garanzia di 90 giorni sulla riparazione effettuata; preventivo di riparazione; diagnosi completa del corretto funzionamento di A500 e A2000; recupero dati da hard disk; upgrade degli apparecchi; verifica della validità della garanzia; archivio storico riparazioni; installazione di hard disk e floppy disk drive.

Computer Lab
Via Cadore, 6
20135 Milano
(Tel. 02/5464436 - Fax 5465036)

INFORSCUOLA '91

A Milano (Palazzo Stelline) si terrà dal 16 al 19 aprile 1991 Inforscuola '91, prima mostra convegno d'informatica, telecomunicazioni, telematica ed editoria per la scuola, promossa dalla ANP (Associazione Nazionale Presidi) e patrocinata dal Ministero della Pubblica Istruzione.

Alla manifestazione parteciperanno presidi, docenti, personale non docente, assessorati all'educazione di comuni, province e regioni di tutta Italia.

Il programma dei convegni è in via di completamento; i temi già definiti

sono: valenza formativa dell'informatica nelle scuole, formazione dei docenti, informatica come ausilio agli studenti portatori di handicap, telecomunicazione per la scuola, telematica per la scuola, software didattico per la scuola, informatica ed editoria scolastica, informatica per l'orientamento dello stuedente, multimedialità nella scuola, dischi ottici nelle tecnologie educative, ruolo delle reti in ambito didattico, gestione scolastica sia didattica sia amministrativa.

Sono previsti inoltre workshop con presentazioni delle esperienze più significative realizzate nelle scuole superiori e dell'obbligo per matematica, italiano, lingue straniere, materie tecniche.

Per informazioni e richieste di adesione rivogersi alla Segreteria Organizzativa della Mostra Convegno:

C.T.I.
Via Palestrina, 31
20124 Milano
(Tel. 02/6691338-6691436
6705390-6705393)

24° SIM HI-FI IVES

Le principali indicazioni emerse dai cinque giorni del SIM '90 (che si è chiuso il 24 settembre alla Fiera di Milano) sono: la conferma del boom dell'home video, la commercializzazione del sistema di registrazione DAT, la registrazione domestica su Compact Disc, un sensibile incremento del car-stereo, l'esasperata ricerca della perfezione nell'hi-fi, la crescente sofisticazione dei videoregistratori, la miniaturizzazione delle telecamere, la "navigazione elettronica" su auto, la diffusione di videoproiettori e televisori ormai vicini ai 40" sempre più stereofonici, digitalizzati, multifunzione, il trionfo dell'interfaccia MIDI nel campo degli strumenti musicali, l'elevata integrazione tra audio e video, l'introduzione crescente del computer per il controllo d'intere orchestre, l'esplosione dei telefoni cellulari portatili e il ritorno massiccio alle console nel campo dei videogiochi.

I visitatori di questa edizione sono stati complessivamente 165.000, di cui 24.500 erano operatori.

UNIX SYSTEM V RELEASE 4

Il boom della domanda di sistemi aperti fa esplodere il mercato per Unix System V Release 4.

Nell'ambito di Unix Expo, la Unix International ha reso noto che quest'anno oltre 300 aziende hanno distribuito prodotti basati sullo Unix System V Release 4, che includono una gamma completa di software e di sistemi avanzati.

Un'indagine ha consentito di censire un totale di 323 produttori che hanno distribuito software basato sulla Release 4, di cui 144 operano in America, 158 in Europa e le altre 21 sono società che raggruppano la maggior parte dei produttori di Asia e Pacifico.

Peter Cunningham, presidente della Unix International ha dichiarato: «Il mercato di System V Release 4 si è sviluppato come previsto. Siamo ormai giunti a una fase decisiva in cui il mercato dopo aver raggiunto una dimensione instabile, oggi assiste all'esplodere della domanda del prodotto. Oggi siamo in grado di puntare verso il mercato mondiale frutto di questa esplosione».

La Unix International è un'organizzazione senza scopo di lucro impegnata nello stabilire le direttive dell'evoluzione del System V Release 4. Costituita nel 1988, i suoi sempre più numerosi associati contano oggi oltre 180 utenti, produttori di software e di sistemi, rivenditori, enti, istituzioni accademiche ed enti governativi in tutto il mondo. La Release 4 raggruppa un mercato di circa 10 milioni di utenti Unix.

Va ricordato che Unix è un marchio registrato della Unix System Laboratories Inc., che è una filiale della AT&T.

Unix International
(Steinar Hoistad - Operations Director)
Avenue de Beaulieu 25
1160 Bruxelles - Belgio
(Tel. 0032/2/6723700)

In Italia contattare:
SCR Associati
(Viviana Villa)
Foro Bonaparte, 70
20121 Milano
(Tel. 02/8026329

ESTERO

CAMPIONATI INTERNAZIONALI DI VIDEOGIOCHI

Durante l'ECES londinese, manifestazione dedicata alle varie forme d'intrattenimento elettronico, svoltasi a metà settembre, si sono tenuti per la prima volta i Campionati Internazionali di Videogiochi.

Al torneo, al quale sono state invitate squadre di quattro elementi provenienti da ogni parte del pianeta, si è particolarmente distinta la nazionale italiana, giunta seconda dietro alla scuderia inglese, distanziando nettamente nelle classifiche finali le altre formazioni. La squadra, composta da giornalisti specializzati fra i quali anche il nostro Fabio Rossi (nominato inoltre "Man of the Show" grazie alle sue videoperformance), parteciperà anche alla prossima edizione della manifestazione, che si terrà in una nazione ancora da stabilire.

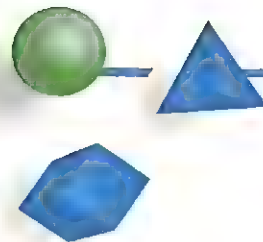
LE NOVITÀ ICD PER L'AMIGA

Flicker Fixer Video è la prima scheda che elimina lo sfarfallio dell'*interlace* sull'intera gamma Amiga: A500, A1000 e A2000. Come monitor, è necessario utilizzare un VGA o un multiscan. La scheda funziona sia in standard televisivo NTSC sia in PAL e il suo prezzo di listino è di 499,95 dollari.

AdSpeed porta la velocità del vostro A500/1000/2000 a 14,3 MHz. L'installazione è semplicissima: è sufficiente sostituire la vostra vecchia CPU con AdSpeed e il gioco è fatto. Il prezzo di questo prodotto è di 349,95 dollari.

Infine, l'ultima novità è la ADSCSI 2080, una scheda per A2000/3000 che combina un'espansione di memoria fino a 8 MB con un controller per hard disk. Il prezzo della scheda è di 279,95 dollari.

ICD Incoporated
1220 Rock Street
Rockford, IL 61101, USA
(Tel 001/815/9682228
Fax 9686888)

Classified

Software

Cerco Wordstar 3.30 per Commodore 128 CP/M. Scrivere a: Giuseppe Vavassori – Via Castellaccia, 3 – 52046 Lucignano (AR) – Tel. 0575/836552.

Software, hardware Amiga. Cerco contatti per scambi ultime novità. Dispongo vasto archivio. Richiedere lista, rispondo a tutti. Luca Morotti – Via Lavino, 239 – 40050 Calderino (BO) – Tel. 051/6762106 – Chiedete anche numero BBS (1200 – 38400 Baud).

Scambio programmi per Amiga. Possego un vasto archivio. Telefonatemi, scrivetemi o inviatemi semplicemente la vostra lista. Rispondo a tutti. Annuncio sempre valido. Marco Caciotti – Via A. Allori, 11 – 50127 Firenze.

Commodore 64/128, cedo tutti i migliori programmi su disco e cassetta. Disponibile vastissima softeca, con ultime novità mondiali. Speciale totocalcio. Assistenza soft-hard, massima serietà, competenza, rapidità. Chiedere lista e informazioni a: Giuliano Cinci – Pian dei Mantellini, 44 – 53100 Siena – Tel 0577/47054.

Vendo, scambio programmi e giochi per Amiga. Arrivi settimanali con circa 2000 titoli a disposizione. Telefonare ore pasti al numero: 0442/331565.

Scambio, vendo, acquisto programmi, giochi, utility per Amiga. Inviare e richiedere liste a: Claudio Benedetto – Via Mercanti, 19 – 10121 Torino – Tel. 011/545823.

Cerco programmi per radioamatori, SWL da utilizzare su Amiga, PC IBM, Sinclair Spectrum. Scrivere a: Massimo Asquini – Via dei Savorelli, 54/D – 00165 Roma.

Scambio, vendita, acquisto programmi per C-128 (modi C-128 e CP/M). Tel. 011/4150675 011/6490116.

Vendo programmi, giochi, ultime novità per C-64/128 su disco o cassetta. Scrivete a: Marilena Gentile – P.zza Padre Pio, Pal. sc. A – 71100 Foggia. Richiedete la mega lista.

Scambio programmi professionali totocalcio per C-64, Amiga, IBM. Max serietà. Onofrio Formisano – Via Venuti, 19 – 80056 Ercolano (NA). Annuncio sempre valido.

Scambio, vendo e compro programmi per Commodore 128. Disponibile vasta biblioteca software. Cerco inoltre un mouse per il suddetto computer, nuovo ma a buon prezzo. Scrivere a: Daniele Poletti – Via Cavo Benedettino, 11 44048 S. Maria Codifiume (FE). Max serietà: rispondo a tutti.

Disponibilità per C-64 di un vastissimo numero di programmi per qualsiasi esigenza. Inoltre programmi professionali per totocalcio, enalotto, totip, lotto. Assistenza soft: massima competenza, esperienza, serietà, rapidità. Rivolgetevi con fiducia a: Luigi Cirillo – Via Ghiaie, 20 – 38100 Trento – Tel. 0461/930500.

Scambio programmi per C-128, sia in modo 128 che CP/M. vasta biblioteca software per questo fantastico e dimenticato computer. Annuncio sempre valido. Inviare liste a: Guido Barolla – C.P. 20 – 55040 Ripa (LU) – Tel. 0584/760693.

Scambio software per Amiga, con manuali. Inoltre cerco contatti per Amiga, zona Trieste. Scrivete a: Dino Chierego – Via della Tesa, 8 – 34137 Trieste o telefonare ore 20-21 al numero 040/392984.

Scambio programmi in MS-DOS. Massima serietà. Inviare liste a: Rosario Albanese – C.so Alberto Amedeo, 66 – 90138 Palermo – Tel. 091/327764.

Scambio e compro programmi per Amiga 500. Annuncio sempre valido. Assicuro risposta. Cristiano Cravanzola, 16 – Via Professor Oliva – 12011 Borgo San Dalmazzo (CN) – Tel. 0171/269659.

CLASSIFIED DELLA COMMODORE GAZETTE È UN MODO ECONOMICO PER INFORMARE LA PIÙ VASTA UTENZA COMMODORE SUI VOSTRI PRODOTTI O SERVIZI. GLI ANNUNCI NON A SCOPO DI LUCRO, INVIATI DA PRIVATI, VENGONO PUBBLICATI GRATUITAMENTE (COMPILARE L'APPOSITA SCHEDA DI SERVIZIO LETTORI).

Quote: 15.000 lire per linea, minimo 4 linee. Aggiungere 5.000 lire per ogni parola in grassetto o 50.000 lire per l'intero annuncio in grassetto.
Condizioni: pagamento anticipato. Vengono accettati assegni e vaglia postali. Gli assegni devono essere intestati a: IHT GRUPPO EDITORIALE s.r.l.
Forma: gli annunci sono soggetti all'approvazione dell'editore e devono essere scritti a macchina o in modo molto chiaro. Una linea equivale a 40 lettere, spazi tra le parole compresi. Pregasi sottolineare le parole che si intendono scrivere in grassetto.
Informazioni generali: gli inserzionisti devono sempre specificare nome e indirizzo completo. Gli annunci appariranno nel primo numero disponibile dopo il ricevimento.

Inviare il materiale a:
IHT GRUPPO EDITORIALE
UFFICI PUBBLICITARI
VIA MONTE NAPOLEONE, 9
20121 MILANO

Attenzione: Commodore Gazette non si assume responsabilità in caso di reclami di qualunque natura da parte degli inserzionisti e/o dei lettori. Nessuna responsabilità è altresì accettata per errori e/o omissioni di qualsiasi tipo.

Vendo in blocco 50 dischetti a L. 50.000 per Commodore C-128 (modo 128 e CP/M). Inoltre scambio programmi per Amiga 500. Scrivere a: Massimo Tabasso – P.zza Molineri, 1 – 12038 Savigliano (CN).

Scambio programmi e giochi per Amiga. Disponibili ultime novità. Massima serietà, assicuro risposta a tutti. Per contatti, telefonare al numero 081/8713539, oppure scrivere a: Silio Scarpone – Via C. Marano, 6 – 80053 C/Mare di Stabia (NA).

Scambio programmi per Amiga 500. Eventualmente acquisto se a buon prezzo. Annuncio sempre valido. Assicuro risposta. Tel. 0171/269659.

Per Amiga metto a disposizione tantissimi programmi di ogni genere. A chiunque ne facesse richiesta invio liste e aggiornamenti. Nessuno scopo di lucro. Scrivere a: Piero Cannizzaro – C.P. 3019 – 50127 Firenze.

Scambio programmi Amiga. Rispondo a tutti. Escludere i venditori. Scrivere a: Claudio Francini – Via delle Gardenie, 82 – 50047 Prato (FI). Annuncio sempre valido.

Scambio software per Amiga. Vasto archivio. Software MS/DOS su disk 5,25" - 3,5" per IBM compatibile. Richiedere lista su disk 3,5" o su carta, allegando affrancatura P.T. a: ARCC – Box 8 – 51030 San Felice (PT).

Scambio, vendo, acquisto programmi per Amiga 500. Ricevo molte novità. Massima serietà. Scrivere e/o telefonare a: Alessandro Brunod – Via Lago S. Michele – 10015 Ivrea (TO) – Tel. 0125/46472 (pom./sera).

Vendo per Amiga 500: gruppo di 50 dischi tra giochi, utility e programmi di P.D., a sole L. 150.000. Inviare vaglia postale a: Andrea Padoan – Via Calvi, 33 – 30175 Marghera (VE).

Scambio software, manuali e idee per Amiga. Scrivere a: Arturo Tonazzi – U.S. Giacomo, 131 – 39050 S. Giacomo di Laives (BZ) – Tel. 0471/940876, ore serali.

Vendo per CBM 64 e Spectrum cassette acquistate in edicola con i migliori programmi, nonché numeri arretrati delle migliori riviste per computer e videogames. Tel. 0564/412940.

Cerco amici con cui scambiare programmi per Amiga senza alcuno scopo di lucro. Scrivere a: Francesco Nasuti – Via G. Vittorio Veneto, 33 - 71043 Manfredonia (FG)

Cerco possessori di Amiga 500 per scam-

bio, vendita, acquisto di software. Possiedo oltre 1000 titoli. Telefonare, solo se interessati, al numero 059/693655, ore pasti. Max serietà.

Vendo programmi per Amiga di ogni genere a basso prezzo. Vendo inoltre 100 dischetti 3,5" garantiti a L. 1.500 l'uno. Telefonare al numero 019/808454 e chiedere di Cristiano. Annuncio sempre valido.

Hardware

Espansioni memoria per Amiga 500/1000/2000 di propria produzione vendo direttamente con garanzia. Per Amiga 1000, espansione memoria aggiuntiva esterna slim line autoconfigurante da 512Kb L. 300.000, da 1Mb L. 400.000, da 2 Mb L. 600.000. Per Amiga 500, da 512K L. 120.000, da 2Mb interna L. 420.000. Per Amiga 2000, da 2 Mb L. 500.000, da 4Mb L. 700.000. I prezzi indicati sono comprensivi di Iva. PC WARE – Tel. 06/6115503.

Amiga 2000, completo di monitor 1084 e filtro polarizzatore, espansione di memoria da 2 Mb (A 2058), 2 drive. Il tutto in ottimo stato, vendo a L. 2.500.000. Claudio Perchinumio – Via Lago d'Iseo, 53 – 36100 Vicenza – Tel. 0444/920116.

Compro C-128D, senza monitor ma con cavi RGB e CVBS + eventuali accessori e/o espansioni + software (linguaggi ed utilities) anche separatamente. Preferibilmente prov. di Salerno. Tel. 0828/370562 ore serali (dopo le 20).

Vendo PC Olivetti M250: microprocessore 80286, HD 40 Mb, 1 drive da 1,44 Mb, 1 Mb di memoria RAM, sistema operativo MS DOS versione 3.30, completo di manuali, monitor monocromatico 12" paper white, tastiera 102 tasti e Stampante Olivetti PR 24 L a 132 colonne, 240 CPS, bidirezionale. Tel. 051/909752 ore serali

Vendo C-64 usato, in ottimo stato + drive + registratore + 1 joystick + 13 cassette, giochi e utility + 8 dischi giochi e utility + 2 manuali d'uso + scrivania di supporto in legno a L. 500.000 trattabili, in zona Novara e provincia. Telefonare, ore serali, al numero 0321/453866.

Vendesi stampante grafica per Amiga, Mannesman Tally 81, acquistata da poco, ancora in garanzia. Usata pochissimo. Completa di imballo, manuale in italiano. Funzionante al 100%. Spedisco in qualsiasi punto d'Italia. Per Taranto fornisco prove di funzionamento. La cedo ad un prezzo insignificante. Alberto Pace – Via Terni, 25 – 74100 Taranto.

Vendo 3 schede hardware Syncro Expert III, esterne, per Amiga 500/1000/2000, ancora da arrivare, causa errato acquisto. Queste schede copiano qualsiasi programma in 45 secondi in qualunque formato (Atari, IBM, Amiga, Machintosh, ecc.)!!! Pagate L. 90.000, cedo per sole L. 39.000!!! Vero Affare! Diego – Tel. 0933/938404.

Attenzione!! Vendo PC Compaq 286e (nuovo) con 1 Mb di RAM, 1 HD da 20 Mb, 1DD da 1,2 Mb, VGA (parallela, seriale, mouse) + monitor bianco e nero VGA + programmi. Tutto a L. 3.000.000 trattabili. Tel. 0771/741263, Raffaele.

Vendo Commodore 128 + disk drive 1571 + stampante Commodore MPS 803 + monitor fosfori verdi + diversi pacchetti software originali (compreso GEOS) a lire 750.000 trattabili. Riccardo Perni – Via Umberto Calosso, 7 – 00155 Roma – Tel. 06/4072737.

Cerco espansione 1750 + espansione video interna da 64K per Commodore 128. Telefonare a: Marco Romani – 06/862551, orario d'ufficio.

Vendo C-64 + disk drive 1541 + stampante MPS 803 + registratore originale + 2 joystick + 60 cassette giochi/utility + 100 giochi in disco + GEOS originale + istruzioni e imballi, a L. 750.000. Fabio Bortolami – Tel. 02/26223736.

Vendo per Amiga 2000 scheda A 2088 XT + floppy disk drive da 3,5" 720 Kb a L. 500.000, con tre mesi di vita. Tel. 051/265448, chiedere di Claudio.

Vendo per A-500 espansione 512K, usata, causa upgrade a 20Mb. Cedo a L. 85.000. Graziano. Tel. 0376/631594 – Mantova. Telefonare dopo le 18.30.

Cedo hardware novità per Amiga: Action Replay, cabinet, cartuccia PC board, drive esterno, track drive, espansioni 512K e/o 2 Mb, hard disk, stampanti, monitor, dischi neutri. Tel. 080/686059.

Vendo espansione originale A 501 – 512K + orologio, in garanzia a L. 100.000 + hard disk A 590, in garanzia a L. 600.000. Telefonare ore 19.30 a Francesco – 0331/597110. Legnano (MI).

Cerco Commodore Vic 20 o C-64 anche in cattive condizioni purché funzionante. Offro rispettivamente L. 80.000 e L. 110.000. Rivolgersi al numero 0377/84334 ore pasti.

Vendo A-2000 1 Mb chip RAM + Exp. 2 Mg Hardital + Genlock ECR Ext. + floppy esterno. Il tutto acquistato 5/90 con imballi e manuali originali a L. 2.000.000. Omaggio 2 joy + prg. Tel. 0932/950972, ore 14-20 solo Sicilia.

Vendo digitalizzatore video e audio stereo per Amiga 500-2000 a L. 195.000. È perfettamente funzionante: supporta grafica in alta e bassa risoluzione e ham, non necessita di alimentazione esterna, è corredato di software dedicato ed è perfettamente compatibile con Audiomaster II. Patrick Ottoz – Telefonare al numero 0771/269232 di pomeriggio dopo le ore 14.00.

Varie

Scambio esperienze e programmi per Macintosh, preferibilmente in formato Amax. Vasta softeca. Inviare lista o telefonare a: Riccardo Solmi – Via Leronia, 49 – 40133 Bologna – Tel. 051/386356 (tardo pomeriggio).

Incontriamoci su Max BBS, la prima BBS a Salerno. Tel. 089/756281 – 300-1200-2400 Baud Bell 103/202 CCITT V21-V22 bis, 25 Megabytes on-line. Orario 22.30-06.30, tutti giorni.

Utente Amiga 2000 con scheda 68020 cerca altri possessori della medesima configurazione per scambio programmi e/o informazioni. Luca Signorelli – Via Fabio Filzi, 15 – 37045 Legnago (VR) – Tel. 0442/21819.

Ormai siamo nell'era delle comunicazioni. Divertiti da 300 a 19,200 BPS 24 ore su 24 ad essere sempre all'avanguardia. Chiama le nostre banche dati amatoriali e non ci crederai... Tel. 02/58106572 – 02/58103714 – 02/57401757 – 02/2827442.

Cerco programmatori assembler per scambio sorgenti, programmi (esclusi giochi), manuali ed informazioni per Amiga. Annuncio sempre valido. Bruno Giuliani – Via Ferdinando Micheli, 26 – 54036 Marina di Carrara (MS).

S.S.I. Amiga Club mette a disposizione i seguenti manuali tradotti da noi (integrali e in italiano): Sculpt 4D, Caligari, Broadcast Videotitler, Animagic, Video Effects 3D e molti altri. Tel. 0835/559053. Specialisti in desktop video.

Cerco documentazione del Professional Page per Amiga. Telefonare nelle ore serali al numero 0427/71649, oppure scrivere a: Michele Beltrame – Via S. Francesco, 16 – 33085 (PN).

Vendo libro "Linguaggio C" 2° edizione di Kernighan e Ritchi (pag. 361) a L. 20.000 Inoltre vendo amplificatore per auto 20W + 20W (Pioneer, modello BP-320) al migliore offerente. Roberto Bolli – Via XI Febbraio 06055 Marsciano (PG).

Cambio autoradio Pioneer Keh 9080 B con Amiga 500 1 Mega, funzionante, oppure vendo L. 550.000. Scrivere a: Marco D'Amico – Via Pio IX, 195 – 00167 Roma. Solo zona limitrofa.

Vendo riviste: Commodore Gazette n. IV 4-5 – V 1-2 – IV 3 a L. 5000, Enigma n. III 5-6-7-8 – IV 4-5-6-7-8-9 L. 5000, Amigabyte n. 16-18-22-24 L. 10.000. Stefano Vitagliano - Via O. Fragnito, 43 – 80131 Napoli – Tel. 081/5453570, ore 9.00 P.m.

Cerco riviste di Commodore Computer Club e altre purché rientrino nell'anno 85-86-87-88-89 a prezzi modici. Spedire a: Michele Franceschi – Via G. Caboto, 7 – 31059 Branco (TV) – Tel. 0422/978586.

Commodore Club

Grosseto Amiga Club offre senza scopo di lucro i migliori programmi, ultime novità USA e pubblico dominio. Iscrizione gratuita. Massima serietà. Tel. 0566/81685, dalle 9.00 alle 17.30.

È nato il C-128 Star! Per saperne di più, scrivi o telefona a: Loris Bontempi – Via Roma, 11 – 46040 Pozzolo (MN) – Tel. 0376/460023.

Iscrivetevi al primo club che invia ogni mese a tutti i soci una rivista con una scelta dei migliori videogiochi e novità per il C-64! Contattateci subito! Scrivete a: Rocco Gregorio – Via Appia, 65 – 85050 Baragiano Sc. (PZ) o telefonate a Piero: 0971/993258

Amiga Club Napoli. Distribuzione gratuita software di pubblico dominio e shareware. Vasta disponibilità di manualistica tecnica. Per informazioni: Massimiliano Afiero – Corso Italia, isolato 11 – 80021 Afragola (NA) – Tel. 081/5527940.

Amiga Club Enna cerca soci. Regala 100 titoli ai nuovi iscritti. Tutte le novità. Consigli per programmatori. Trucchi e soluzioni dei giochi. Telefonate al numero 0935/22658.

Detection Mode World Users è l'unico club che ti permette di sapere e avere tutto prima di tutti. Biblioteca programmi illimitata con novità in arrivo tempestivo. Inoltre, manuali, libri, tricks, intro. Chiamaci al numero 0825/71710 o scrivi a Danny – Via De Meo, 2 – 83100 Avellino.

È nato a Salerno da circa due anni un favoloso computer club. Il suo nome è "Computer Eyes Salerno". Noi cerchiamo nuovi soci sia per scambio (ultime novità) sia per eventuale adesione. Telefonateci ore pasti ai seguenti numeri: 089/332106 (Massimo) e 089/723296 (Alessandro).

Quinta Dimensione Italy. Disponibile qualsiasi programma e manuale per Amiga. Servizio abbonamenti con condizioni particolari per club. Arrivi giornalieri. Per informazioni, telefonare al numero 0362/501857. Luca, ore 19.00-21.00.

Il Luna User Group offre a tutti i soci hardware a prezzi scontatissimi: dischi Bulk da L. 650, espansione 512K (A500) da L. 90.000, espansione 2Mb da L. 340.000. Tutto in garanzia. Nazzareno Gorni – Via Roma, 8 – 26034 Piadena (CR) – Tel. 0375/98745.

Il Visoft Club Palermo mette a disposizione dei soci una vastissima biblioteca software senza scopo di lucro; si scambia software. Telefonaci al numero 091/8905469 o scrivi a: Visoft Incardona Salvatore – Via Di Stefano, 109 – 90047 Partinico (PA).

Softgallo Mega Amiga Club, cerca contatti in tutta Italia per scambio di opinioni ed eventualmente programmi; archivio di oltre 10.000 programmi in aggiornamento giornaliero. Scrivere a: Softgallo Team! P.zza Buonarroti, 25 – 20149 Milano. ■

INDICE DEGLI INSERZIONISTI

Direzione vendite spazi pubblicitari:

IHT Gruppo Editoriale - Commodore Gazette
Agente Pubblicitario: Aldo Pagano Pagano
Via Monte Napoleone, 9 - 20121 Milano
Tel. 02/794181 - 799492 - 792612 - 794122
Telex 334261 IHT I - Telefax 02/784021

Questo indice è da considerarsi come un servizio addizionale. L'Editore non si assume alcuna responsabilità per eventuali errori e/o omissioni.

Indirizzare eventuali lamentele riguardanti gli inserzionisti a:

Commodore Gazette
Uffici Pubblicitari
Relazioni Inserzionisti
Via Monte Napoleone, 9 – 20121 Milano

Nessuna responsabilità viene altresì assunta dalla Commodore Gazette per eventuali problemi di qualsiasi natura con gli inserzionisti. La responsabilità di quanto pubblicato negli spazi pubblicitari è esclusivamente del committente.

COME DIGITARE I LISTATI DI COMMODORE GAZETTE

I listati per C-64/128 contengono una particolare simbologia. Tutti i caratteri grafici e quelli di controllo sono stati tradotti in combinazioni di tasti facilmente comprensibili. Sono le istruzioni tra parentesi graffe. Per esempio, {SHFT L} indica che si deve tener premuto il tasto shift e premere una volta il tasto L. Ovviamente, non bisogna digitare le parentesi; quello che apparirà sullo schermo saranno simboli grafici. Altri esempi:

{20 SPAZI} premere la barra spaziatrice 20 volte.
{SHFT CLR} tenere premuto il tasto shift e premere una volta il tasto clr-home.
{2 CRSR ↓} premere cursore-giù due volte.
{CTRL 1} tenere premuto il tasto control e premere il tasto 1.
{COMD T} tenere premuto il tasto col logo Commodore e premere T.
{CRSR ←} premere cursore-sinistra una sola volta.
{SHFT A} tenere premuto il tasto shift e premere il tasto A.

Gli altri tasti che non danno origine a caratteri particolari (come ↑, £, @) sono invece presentati normalmente.

Manoscritti: e collaborazioni dei lettori – manoscritti, disegni e/o fotografie – sono benvenute e verranno valutate in vista di una possibile pubblicazione. Commodore Gazette non si assume comunque responsabilità per perdite o danni al materiale. Si prega di allegare una busta affrancata e indirizzata per ogni articolo. Il pagamento per materiale non richiesto viene effettuato solo in seguito all'accettazione da parte della redazione. I contributi editoriali di qualunque forma non si restituiscono. Tutta la corrispondenza editoriale, richieste di annunci, problemi di sottoscrizione abbonamenti, di diffusione e con gli inserzionisti, deve essere indirizzata a: Commodore Gazette – Ufficio Editoriale – Via Monte Napoleone 9 – 20121 Milano. **Commodore Gazette** è un periodico indipendente non connesso in alcun modo con la Commodore Business Machines e con tutte le sue sussidiarie e affiliate, compresa la Commodore Italiana S.p.A. Commodore Gazette viene pubblicata dalla IHT Gruppo Editoriale, Via Monte Napoleone 9, 20121 Milano. Nessuna parte di questa pubblicazione può essere in alcun modo riprodotta senza permesso scritto dell'editore. La redazione si adopera per fornire la massima accuratezza negli articoli e nei listati pubblicati. Commodore Gazette non si assume responsabilità per eventuali danni dovuti a errori od omissioni.

IL PROSSIMO NUMERO SARÀ IN EDICOLA ALLA FINE DI DICEMBRE

SERVIZIO LETTORI

Questa scheda è valida fino al 31 dicembre 1990

A. Come giudica questo numero di Commodore Gazette?

- ☐ 1. Ottimo
- ☐ 2. Molto buono
- ☐ 3. Buono
- ☐ 4. Discreto
- ☐ 5. Sufficiente
- ☐ 6. Mediocre
- ☐ 7. Insufficiente

B. Quale(i) articolo(i) di questo numero ha apprezzato maggiormente?___

C. Quale(i) articolo(i) di questo numero giudica peggiore(i)?___

D. Quali argomenti dovrebbero essere trattati nei prossimi numeri di Commodore Gazette?___

E. Con quale aggettivo descriverebbe Commodore Gazette?___

F. Quante persone leggono la sua copia di Commodore Gazette?

- ☐ 1. Uno
- ☐ 2. Due
- ☐ 3. Tre
- ☐ 4. Quattro o più

G. Ha dei suggerimenti?___

H. Quale(i) computer utilizza?

- ☐ 1. C-64
- ☐ 2. C-128
- ☐ 3. C-128D
- ☐ 4. Amiga 500
- ☐ 5. Amiga 1000
- ☐ 6. Amiga 2000
- ☐ 7. Altro (specificare)___

I. Quale(i) computer intende acquistare nel futuro?

- ☐ 1. C-64
- ☐ 2. C-128D
- ☐ 3. Amiga 500
- ☐ 4. Amiga 2000
- ☐ 5. Altro (specificare)___

L. È un acquirente dei libri della IHT? Se sì come li giudica?___

M. Quali altre riviste (sia d'informatica che non) legge abitualmente?___

N. Indichi quali sono i suoi maggiori interessi

- ☐ 1. Videoregistrazione
- ☐ 2. Hi-Fi
- ☐ 3. Strumenti musicali
- ☐ 4. Fotografia
- ☐ 5. Automobili
- ☐ 6. Sport
- ☐ 7. Viaggi

O. Quali periferiche intende acquistare nei prossimi sei mesi?___

P. Quanto intende spendere in software e hardware nei prossimi sei mesi?___

Nome ___
Cognome ___
Indirizzo ___
Città ___
Prov. ___ **C.a.p.** ___ **Età** ___
Professione ___

COMMODORE GAZETTE **novembre 1990**

SCHEDA ORDINAZIONE LIBRI

Con il presente tagliando desidero ordinare i(l) seguenti(e) libri(o):

Collana Informatica

- ☐ L'Amiga (Michael Boom) .... L. 60.000
- ☐ Il Manuale dell'AmigaDOS (Commodore-Amiga) .... L. 60.000
- ☐ Programmare l'Amiga Vol. I (Eugene P. Mortimore) .... L. 80.000
- ☐ Programmare l'Amiga Vol. II (Eugene P. Mortimore) .... L. 70.000
- ☐ Il Manuale dell'hardware dell'Amiga (Commodore-Amiga) .... L. 76.000
- ☐ Guida ufficiale alla programmazione di GEOS (Berkeley Softworks) .... L. 64.000
- ☐ Flight Simulator Co-pilot (Charles Gulick) .... L. 30.000
- ☐ Volare con Flight Simulator (Charles Gulick) .... L. 45.000

Collana Cinema

- ☐ Le mille luci di Hollywood (David Chell) .... L. 42.000

Collana Tempus

- ☐ Inventori del nostro tempo (Kenneth A. Brown) .... L. 42.000
- ☐ Computer in guerra: funzioneranno? (David Bellin e Gary Chapman) .... L. 39.900
- ☐ La sfida della crescita (G. Ray Funkhouser e Robert R. Rothberg) .... L. 39.900
- ☐ La Macchina e la Mente (George Johnson) .... L. 42.000
- ☐ I Creatori del Domani (Grant Fjermedal) .... L. 39.900

Pagherò in contrassegno al postino la somma di L........ + spese postali (L. 6.000 per volume)

Nome e cognome ___
Indirizzo ___
Città ___
Prov. ___ **C.a.p.** ___ **Tel.** ___
Firma ___

novembre 1990

☐ Desidero inserire gratuitamente un mio annuncio nella rubrica CLASSIFIED (solo per i privati e per gli annunci non a scopo di lucro).

☐ Desidero inserire un mio annuncio nella rubrica CLASSIFIED. Allego assegno bancario o circolare o postale oppure fotocopia della ricevuta del vaglia postale per un totale di L. ______________________ . Il mio codice fiscale o partita IVA (per le aziende) è il seguente:______________________________________.

Attenzione: perché un annuncio venga accettato è necessario che sia stato compilato anche il questionario presente sull'altro lato di questo tagliando. Non si accettano fotocopie.

TESTO:____
____
____
____

Inserire all'interno di una busta affrancata e spedire a:

Commodore Gazette
Servizio Lettori
Via Monte Napoleone, 9
20121 Milano

- - - - - ✂ -

Inserire all'interno di una busta affrancata e spedire a:

IHT Gruppo Editoriale
Divisione Libri
Via Monte Napoleone, 9
20121 Milano

IL MANUALE
DELL'HARDWARE
DELL'AMIGA

COMMODORE-AMIGA, INC.

AMIGA

Il volume è l'edizione in lingua italiana del primo dei tre manuali appartenenti alla libreria di riferimento tecnico dell'Amiga, realizzati dalla stessa azienda produttrice, la Commodore-Amiga. Questo manuale è una fonte d'informazioni ufficiale, uno strumento di riferimento essenziale per tutti i programmatori in linguaggio Assembly che hanno bisogno d'interagire con la macchina in maniera diretta; per i progettisti che intendono creare nuove periferiche per l'Amiga; per chiunque sia interessato a scoprire come funziona l'hardware dell'Amiga.

Gli argomenti principali sono: l'hardware del Copper, dei playfield, degli sprite, audio, del Blitter, di controllo e d'interfaccia. Non mancano delle utili appendici (registri, mappa di memoria, connettori, interfacciamento) e un glossario.

336 pagine - 18,8 x 23,5 cm - ISBN 88-7803-018-X - Lire 76.000

I libri IHT sono disponibili nelle migliori librerie e computer shop. Per ordini diretti servirsi dell'apposito modulo pubblicato a pagina 95

IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-794122 - Fax 784021 - Telex 334261 IHT I

Distribuzione: RCS Rizzoli Libri - Via Scarsellini, 17 - 20161 Milano - Tel. 02/64068508